NAVAL POSTGRADUATE SCHOOL

MONTEREY, CALIFORNIA

THESIS

CYBER EVENT ARTIFACT INVESTIGATION TRAINING IN A VIRTUAL ENVIRONMENT

by

Simone M. Mims
Tye R. Wylkynsone

December 2017

Thesis Advisor:	J.D. Fulp
Second Reader:	Gurminder Singh

Approved for public release. Distribution is unlimited.

THIS PAGE INTENTIONALLY LEFT BLANK

REPORT DOCUMENTATION PAGE			Form Approved OMB No. 0704-0188

Public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instruction, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Washington headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302, and to the Office of Management and Budget, Paperwork Reduction Project (0704-0188) Washington, DC 20503.

1. AGENCY USE ONLY *(Leave blank)*	2. REPORT DATE December 2017	3. REPORT TYPE AND DATES COVERED Master's thesis

4. TITLE AND SUBTITLE CYBER EVENT ARTIFACT INVESTIGATION TRAINING IN A VIRTUAL ENVIRONMENT	5. FUNDING NUMBERS
6. AUTHOR(S) Simone M Mims and Tye R. Wylkynsone	
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) Naval Postgraduate School Monterey, CA 93943-5000	8. PERFORMING ORGANIZATION REPORT NUMBER
9. SPONSORING /MONITORING AGENCY NAME(S) AND ADDRESS(ES) N/A	10. SPONSORING / MONITORING AGENCY REPORT NUMBER

11. SUPPLEMENTARY NOTES The views expressed in this thesis are those of the author and do not reflect the official policy or position of the Department of Defense or the U.S. Government. IRB number ____N/A____.

12a. DISTRIBUTION / AVAILABILITY STATEMENT Approved for public release. Distribution is unlimited.	12b. DISTRIBUTION CODE

13. ABSTRACT (maximum 200 words)

The Internet has created many new technology advances that make everyday life easier and more efficient. However, technology has also enabled new attack capabilities and platforms that have the potential to cripple Department of Defense (DOD) and civilian information systems and cyber infrastructure. In order to minimize damages these threats could cause, the DOD needs well-trained operators and skilled cyber incident first responders at the helm. The first portion of this research focused on identifying operating system artifacts that give first responders the best information with which to identify if a cyber incident has occurred, or is occurring, and to determine the type of incident.

The second portion of this research focused on developing virtual environments where students can participate in guided training and challenge labs. These labs can train system operators to recognize incident indicators and allow first responders to focus on collecting necessary information quickly. The Training Lab focuses on leading the student through an investigation of each designated artifact, while the Challenge Lab provides less guidance in order to test the students' acquired skills. This partnered learning experience should lead to more proficient cyber incident reporting and should decrease the response delay between detection and recovery.

14. SUBJECT TERMS cyber event investigation, CIRCE, operating system artifacts, incident first responder, CJCSM 6501.01b	15. NUMBER OF PAGES 119
	16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT	18. SECURITY CLASSIFICATION OF THIS PAGE	19. SECURITY CLASSIFICATION OF ABSTRACT	20. LIMITATION OF ABSTRACT
Unclassified	Unclassified	Unclassified	UU

NSN 7540-01-280-5500 Standard Form 298 (Rev. 2-89) Prescribed by ANSI Std. 239-18

THIS PAGE INTENTIONALLY LEFT BLANK

Approved for public release. Distribution is unlimited.

CYBER EVENT ARTIFACT INVESTIGATION TRAINING IN A VIRTUAL ENVIRONMENT

Simone M. Mims
Lieutenant, United States Navy
B.S., Tulane University, 2011

Tye R. Wylkynsone
Lieutenant, United States Navy
B.S., Marquette University, 2001

Submitted in partial fulfillment of the
requirements for the degree of

MASTER OF SCIENCE IN COMPUTER SCIENCE

from the

NAVAL POSTGRADUATE SCHOOL
December 2017

Approved by: J.D. Fulp
Thesis Advisor

Gurminder Singh, Ph.D.
Second Reader

Peter Denning, Ph.D.
Chair, Department of Computer Science

THIS PAGE INTENTIONALLY LEFT BLANK

ABSTRACT

The Internet has created many new technology advances that make everyday life easier and more efficient. However, technology has also enabled new attack capabilities and platforms that have the potential to cripple Department of Defense (DOD) and civilian information systems and cyber infrastructure. In order to minimize damages these threats could cause, the DOD needs well-trained operators and skilled cyber incident first responders at the helm. The first portion of this research focused on identifying operating system artifacts that give first responders the best information with which to identify if a cyber incident has occurred, or is occurring, and to determine the type of incident.

The second portion of this research focused on developing virtual environments where students can participate in guided training and challenge labs. These labs can train system operators to recognize incident indicators and allow first responders to focus on collecting necessary information quickly. The Training Lab focuses on leading the student through an investigation of each designated artifact, while the Challenge Lab provides less guidance in order to test the students' acquired skills. This partnered learning experience should lead to more proficient cyber incident reporting and should decrease the response delay between detection and recovery.

THIS PAGE INTENTIONALLY LEFT BLANK

TABLE OF CONTENTS

I. INTRODUCTION 1
A. CJCSM FRAMEWORK 2
1. CJCSM Phases 2
2. Research Focus: Detection and Preliminary Analysis Phases 4
3. Cyber Incident Report Format 6
B. OPERATING SYSTEM ARTIFACTS 8

II. VIRTUAL ENVIRONMENT 13
A. SETUP 13
B. EXPLOITATION 14
1. Training Lab 14
2. Challenge Lab 16
C. TRANSFER TO PORTABLE FORMAT 21

III. INVESTIGATION METHODOLOGY FOR ARTIFACTS 23
A. PROCESSES 24
1. Artifact Description 24
2. Investigation Tools 25
3. Evaluation of the Indicators to Determine if Artifact Is Malicious 26
4. Transition Signals 28
B. USERS LOGGED-ON 28
1. Artifact Description 28
2. Investigation Tools 29
3. Evaluation of the Indicators to Determine if Artifact Is Malicious 29
4. Transition Signals 30
C. SCHEDULED TASKS 30
1. Artifact Description 30
2. Investigating Tools 31
3. Evaluation of the Indicators to Determine if Artifact Is Malicious 32
4. Transition Signals 33
D. ACCOUNTS 34
1. Artifact Description 34
2. Investigation Tools 35

3. Evaluation of the Indicators to Determine if Artifact Is Malicious 35
4. Transition Signals 36
E. REGISTRY 37
1. Artifact Description 37
2. Investigation Tools 38
3. Evaluation of the Indicators to Determine if Artifact Is Malicious 39
4. Transition Signals 39
F. FILES 40
1. Artifact Description 40
2. Investigation Tools 42
3. Evaluation of the Indicators to Determine if Artifact Is Malicious 43
4. Transition Signals 44
G. LOGS 44
1. Artifact Description 44
2. Investigation Tools 45
3. Evaluation of the Indicators to Determine if Artifact Is Malicious 46
4. Transition Signals 48
H. NETWORK CONNECTIONS 48
1. Artifact Description 48
2. Investigation Tools 49
3. Evaluation of the Indicators to Determine if Artifact Is Malicious 49
4. Transition Signals 51

IV. CONCLUSION 53
A. SUMMARY 53
B. FUTURE WORK 54
1. Classroom and Individual Use Evaluations 54
2. Initial Response Actions 54
3. Development of Lab in a Game-Like Environment 55
4. Mobile Application and Web Browser Artifacts 55
5. Employ Later Windows OS Versions in the Virtual Environment 55

APPENDIX A. TRAINING LAB 57

APPENDIX B. TRAINING LAB REPORT 77

APPENDIX C. CHALLENGE LAB 81

APPENDIX D. CHALLENGE LAB REPORT 89
A. ATTACKER'S STORYLINE 92
B. ACTIONS BY ARTIFACT 92

LIST OF REFERENCES 95

INITIAL DISTRIBUTION LIST 99

THIS PAGE INTENTIONALLY LEFT BLANK

LIST OF FIGURES

Figure 1. CJCSM Incident Management Process Life Cycle Model. Source: [7]. 3

Figure 2. Operating System Usage. Adapted from [8]. 9

Figure 3. Process Monitor 3 Process Tree View 26

Figure 4. Network Connection Screenshot 51

THIS PAGE INTENTIONALLY LEFT BLANK

LIST OF TABLES

Table 1. DOD Incident and Event Category. Adapted from [6]. 6

Table 2. Initial Impact Matrix. Adapted from [6]. 6

Table 3. Suggested Artifacts Comparison Chart.....11

THIS PAGE INTENTIONALLY LEFT BLANK

LIST OF ACRONYMS AND ABBREVIATIONS

CIRCE	Cyber Incident or Reportable Cyber Event
CJCSM	Chairman of the Joint Chiefs of Staff Manual
CPU	Central Processing Unit
DLL	Dynamically Linked Library
GUI	Graphic user interface
IS	Information System
LNK	Link File
MFT	Master File Table
ObjectID	Object Identifier
OS	Operating System
RFC	Request For Comments
SLsmtp	Seattle Lab simple mail transfer protocol
SVC	Service
US-CERT	United States–Computer Emergency Readiness Team
WinOS	Windows operating system

THIS PAGE INTENTIONALLY LEFT BLANK

ACKNOWLEDGMENTS

We give honor to the invaluable help of God; Tye's wife, Stephanie; and our advisor, JD Fulp, who helped us remain organized and on track throughout this entire process. It is through great assistance from others that this endeavor is finished. Our heartfelt thanks go out to you.

THIS PAGE INTENTIONALLY LEFT BLANK

I. INTRODUCTION

The United States Computer Emergency Readiness Team (US-CERT) defines an incident as "[the violation of an] explicit or implied security policy. This can include but is not limited to attempts to gain unauthorized access, unwanted disruption or denial of service, unauthorized use of a system for processing, or changes to system hardware or software without the owner's instruction or consent" [1]. According to the Office of Management and Budget's annual report to Congress, a report directed by the Federal Information Security Modernization Act of 2014, federal agencies reported over 69,851 cyber incidents in FY14, 77,483 in FY15, and 30,899 in FY16 [2], [3]. Such a sizeable number of attacks per year presents a significant threat to the cyber architecture critical to the daily operations of Department of Defense (DOD), other government, and civilian enterprises. Though the fiscal year 2016 figures show that these attacks have decreased, there are still a significant number of attacks threatening these information systems (ISs). In order to minimize the damage these attacks could cause to United States ISs, system operators should be trained to recognize indicators that a cyber attack was either attempted or successfully committed. The first responder—or the first person notified of and who reacts to a suspicious event [4]—should possess the expertise to investigate and validate the indicator(s), and collect the necessary information needed to make a concise, informative, and timely initial report. This research developed hands-on lab experiences intended to increase an operator's ability to recognize, categorize, and validate (or invalidate) computer-based and network-based indicators through analysis of readily available operating system artifacts [5], [6].

Within an organization, the individuals who interact with information systems fall into one of two general incident-handling roles: that of a detector or that of a responder. The Chairman of the Joint Chiefs of Staff Manual 6510.01B (subsequently referred to in this document as simply "CJCSM") defines detectors as "people who observe an event or incident and are trained to step away from the affected system in order to ensure no damage or contamination of evidence" [6]. A detector is most likely the individual who first notices the indicators of a potential cyber event. These indicators can come from,

among other things, automated protection systems (i.e., alerts), user observation of system abnormalities, or external reports. Responders, on the other hand, are described as "the trained personnel who arrive to investigate and respond to a cyber-event or incident" [6]. A first responder is "the person who is designated within an organization to handle security incidents and determine their root cause" [4]. This research has developed hands-on lab experience intended to enhance a detector's ability to recognize one or more system events that are indicative of an incident, or are otherwise worthy of reporting. This experience should also train the first responder to more effectively validate and report incidents utilizing available operating system analysis tools and techniques.

A. CJCSM FRAMEWORK

The CJCSM 6510.01B was written and distributed in July of 2012. It focuses on establishing the guidelines, major processes, and required actions for the Department of Defense Cyber Incident Handling Program. The purpose of this program is to "ensure an integrated capability to continually improve the Department of Defense's ability to rapidly identify and respond to cyber incidents that adversely affect DOD information networks and information systems" [6]. This program not only addresses cyber incidents (which convey a direct implication of malice), but also any cyber-related events that do not rise to the severity level of an incident, but nonetheless are useful in gaining and maintaining situational awareness regarding potential threats and vulnerabilities. For brevity, cyber incidents and reportable cyber events will be referred to as CIRCE throughout the remainder of this document.

1. CJCSM Phases

The Cyber Incident Handling program is broken down into the following six phases:

1. Detection of event.
2. Preliminary analysis and identification of incidents.
3. Preliminary response actions.
4. Incident analysis.

5. Response and recovery.

6. Post-incident analysis. [6]

As depicted by Figure 1, these 6 phases come together to serve as a framework encompassing all activities taken to promote prompt detection, timely reporting, and effective containment and recovery of an incident. This framework also facilitates damage minimization on information systems and networks, incident data preservation, coordination and communication across organizations, compilation of lessons learned, and the improvement of current defenses and strategies through pattern analysis [6].

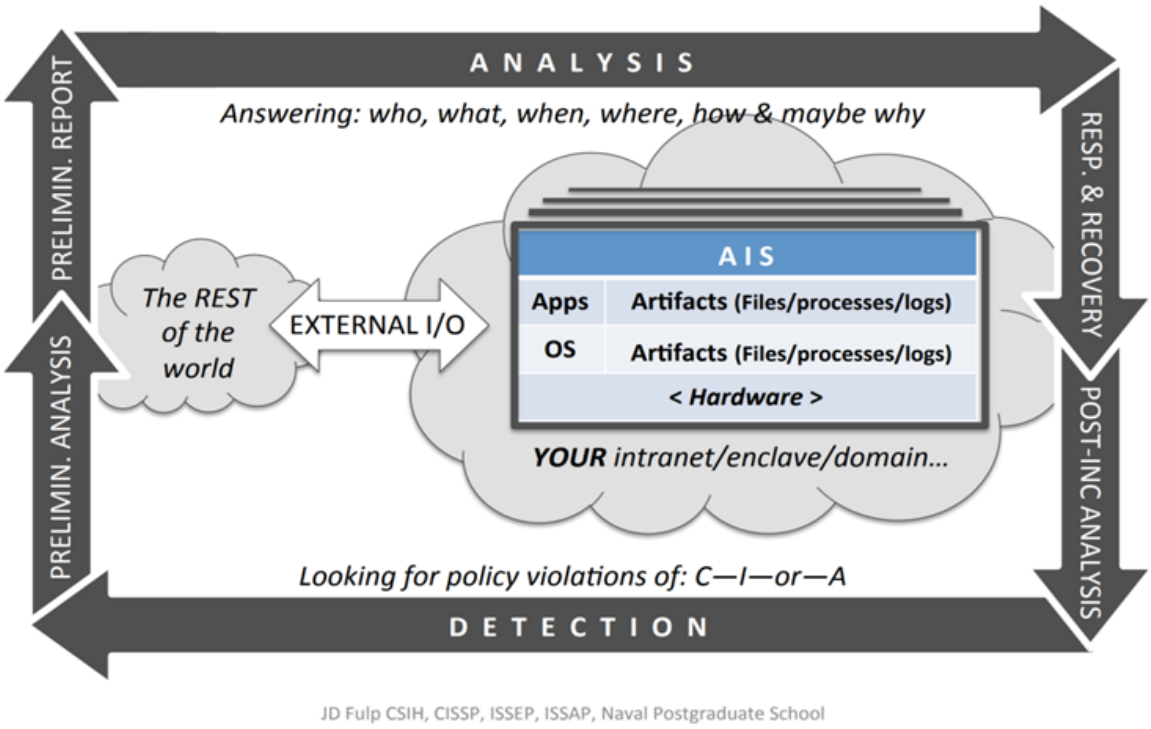


Figure 1. CJCSM Incident Management Process Life Cycle Model. Source: [7].

2. Research Focus: Detection and Preliminary Analysis Phases

The goal of the Detection Phase is to recognize all unusual activity, or events, that have the potential to degrade or harm the network or information system [6]. An organization should continually be in the Detection Phase of the lifecycle; as new CIRCE can occur at any time. Detection may be simultaneous with other ongoing investigations or processing no matter which phase they are in.

Detection is critical. Neither the investigation nor the mitigation of an incident's effects can even begin until a CIRCE has been detected. An effective detection capability within an organization is a direct result of guiding principles regarding what events are considered "normal" during the day-to-day operations of the system. By inference, then, any events falling outside of such a "normal" baseline should be subject to scrutiny. Such guidance should also be accompanied by appropriate training for the personnel assigned the tasks associated with the ongoing system event monitoring that is needed to detect deviations from "normal." This training should include instruction on how to configure the monitoring tools/systems, as well as providing operator-level familiarity with such tools and systems.

Detection of suspicious events can occur in multiple ways. Three methods outlined in the CJCSM 6501.01B are:

1. An automated detection system or sensor.
2. A report from an individual or user.
3. An incident report or situational awareness update from other internal or external organizational components, such as USCYBERCOM or US-CERT. [6]

After detecting a suspicious event, first responders should make an initial notification to the appropriate external entities. The CJCSM 6501.01B, NIST Special Publication 800–61, as well as local written guidance are a few sources that can be utilized to guide proper notification procedures. The precise format of such notification will, in part, be driven by the classification of the system affected, the type of event that was detected, and the particular affected organization security policies in place. Cross-domain coordination is key to ensuring all potentially affected entities can address the

issues as early on as possible. “An event cannot be determined to be an incident until some preliminary analysis is done to assess and validate the event against the criteria for determining if it is an event” [6]. Thus, first responders must move to the Preliminary Analysis Phase of the Incident Handling Model.

In the Preliminary Analysis Phase, a first responder will conduct an initial analysis to determine whether the reported cyber event should be considered a CIRCE [6]. The first responder will review the following information, when available, and compare it to the organization’s incident criteria to make this determination:

1. General description of the problem, event, or activity.
2. Status (ongoing or ended; successful or unsuccessful).
3. Number of ISs affected.
4. Source and destination Internet Protocol (IP) addresses.
5. Source and destination ports.
6. Hostname(s).
7. IS location.
8. User Information.
9. Timestamps.
10. IDS alert and payload data (if relevant). [6]

Once the event is validated as a CIRCE, the first responder will assign that event to an incident category (Table 1), assign an initial impact assessment (Table 2), and begin or continue incident documentation. This documentation will include all known information about the incident and a detailed record of actions taken by the first responder. This will be helpful should the incident transfer to another responder for further analysis or investigation. During the Preliminary Analysis Phase, first responders will also make a determination if computer forensics should be conducted, and submit an initial report in accordance with CJCSM and NIST reporting format [5], [6].

Table 1. DOD Incident and Event Category. Adapted from [6].

Category	Description
0	Training and Exercises
1	Root Level Intrusion (Incident)
2	User Level Intrusion (Incident)
4	Denial of Service (Incident)
7	Malicious Logic (Incident)
3	Unsuccessful Activity Attempt (Event)
5	Non-Compliance Activity(Event)
6	Reconnaissance (Event)
8	Investigating (Event)
9	Explained Anomaly (Event)

Table 2. Initial Impact Matrix. Adapted from [6].

	Cyber Incident and Reportable Cyber Event Category						
Network Device	CAT 1	CAT 2	CAT 3	CAT 4	CAT 5	CAT 6	CAT 7
Backbone	High	High	Low	High	Low	Low	Low
Router	High	High	Low	High	Moderate	Low	Low
Network Management/ Security Server	High	High	Low	High	Moderate	Low	Moderate
Non-Public Server	Moderate	Moderate	Low	Moderate	Moderate	Low	Moderate
Public Server	Low	Low	Low	Moderate	Low	Low	Moderate
Workstation	Low	Low	Low	Moderate	Low	Low	Moderate

3. Cyber Incident Report Format

The CJCSM 6501.01B provides a standardized reporting format in order to ensure that incident reports are standardized across domains. Reports should be as accurate and meaningful as possible. They can be updated as the incident progresses and new or better (i.e., more accurate or detailed) information becomes available [6]. This information will flow simultaneously along two paths: the technical reporting channel and the operational reporting channel. Information that flows along the technical reporting channel will focus on actions taken to mitigate the CIRCE; while the operational channel keeps the chain of

command informed of the status of the event and the impact of the event on any current operations [6].

The Cyber Incident Report is broken down into eight sections:

1. Cyber Incident Tracking Information
2. Reporting Information
3. Categorization Information
4. Technical Details
5. Sites Involved
6. Impact Assessment
7. Additional Reporting
8. Other

During Phases 2 and 3, a first responder should focus on generating a "quick" summary report rather than a complete and detailed report, which would be more appropriate following Phase 4. Given the "quick summary" nature of this early (phase) report, a responder should focus his/her initial efforts on information required for the following report sections:

1. Reporting Information,
2. Categorization Information,
3. Technical Details, and
4. Impact assessment.

This provides the information needed to inform the correct personnel in the chain of command and get the right response actions in motion. Provided below is a brief description of what information should be placed into each relevant section of the report during the preliminary investigation. More detailed guidance is provided in Appendix B to Enclosure C of the CJCSM 6510.01B [6].

a. Reporting Information

This portion of the Cyber Incident Report provides the reader with information about the reporting unit. It also shows how to reach the organization's point of contact for further information on the incident.

b. Categorization Information

This portion of the Cyber Incident Report provides the status of the incident or reportable event as well as the suspected delivery vector and system vulnerability that facilitated the incident. This section would include any actions taken by the first responder to analyze or mitigate the incident.

c. Technical Details

This section includes a description of the event or incident, as detailed as possible, as well as any identifying information gathered on who conducted the attack or from where the attack came. This information can include, but is not limited to, any identifying attributes of the attacking system, such as source socket pairs, country of origin, operating system, and exploits or tools used. This section should also include a list of any targeted IP and ports, the method of detection and a root cause determination, if known.

d. Impact Assessment

A first responder can complete an initial impact assessment even with a limited amount of information concerning the event [6]. This assessment should focus on giving the affected chain of command a snapshot of the operational effect of the incident. Information that can be included here, if known during Phase 2, includes affected systems, operational or technical impact and any known lost work hours.

B. OPERATING SYSTEM ARTIFACTS

Though no official definition exists, one may consider an operating system artifact to be any object or observable phenomenon that one may utilize to understand what has taken place on a system or network. These artifacts often comprise key "evidence" useful in discovering and understanding the pertinent details of what

malicious events/activities may have occurred on a particular system. Each operating system produces its own unique collection of system artifacts. Despite this, the artifacts of all operating systems can be generalized to just a few common types. This research focused on Windows systems and its particular collection of artifacts that are relatively easy for a responder to review. As shown in Figure 2, Windows is one of the most widely utilized desktop/laptop operating systems worldwide.

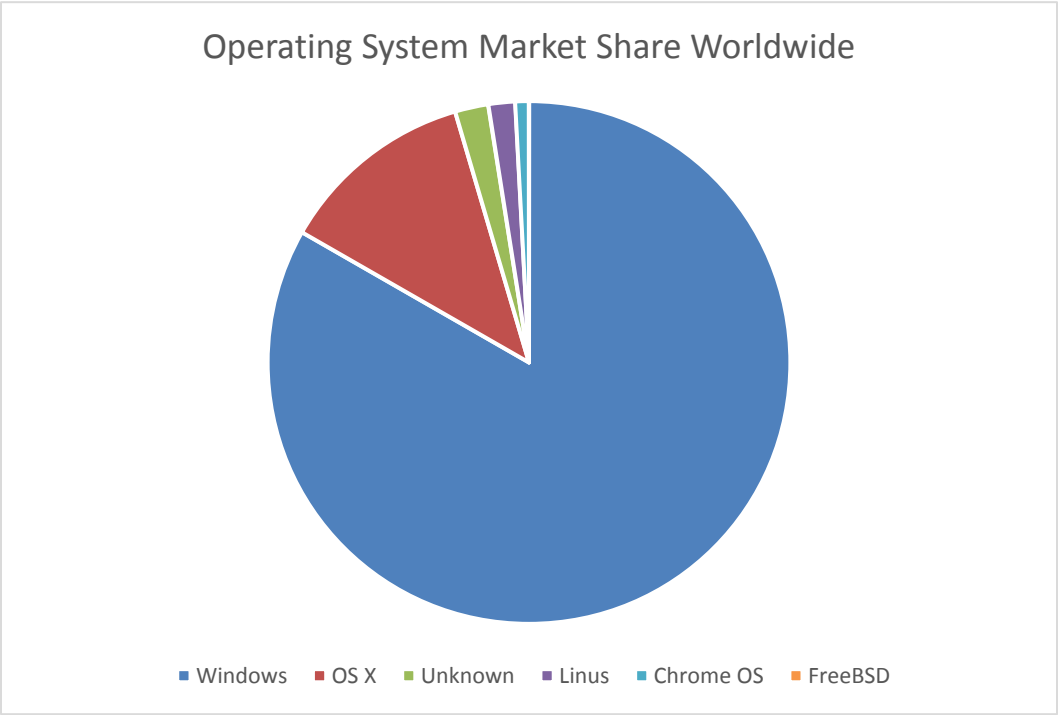


Figure 2. Operating System Usage. Adapted from [8].

Due to the widespread use of Windows computers, it is highly likely that an incident detector or responder will encounter a Windows system that will require digital investigation. Windows systems produce many artifacts. The virtual machine environments created for both the Training and Challenge Labs associated with this research employ the Windows XP operating system. Windows XP was the first Windows operating system (WinOS) to utilize the Windows NT (New Technology) file system. Though now one of the older WinOSs available, many of its basic characteristics are carried over into the newer WinOSs. The "[Windows NT] file system allowed for more

in-depth analysis of the system" [9]. XP was the first version of the Windows operating system to utilize a prefetch file to improve the user experience. Since 2001, this prefetch file has been a pertinent source of information concerning a user's normal computer habits [9]. XP also introduced logging functionality into the Windows operating system [10]. All subsequent versions of Windows are built upon these XP native functions. For these reasons, this research utilized the XP WinOS to assist students in developing a strong foundation for analysis that reasonably extrapolates to later Windows operating systems.

This research suggests the following list of artifacts be considered for analysis during the Preliminary Analysis Phase of the Incident Handling Cycle. This list is based on the overlap of artifacts between the various artifact sources listed in Table 3, along with the emphasis of ensuring a timely and informative report. Due to this latter emphasis, we avoided any analysis method or technique that was advanced, specialized, or otherwise so complex that it would likely exceed the skills of the typical first responder, or take so long to accomplish that it would hinder a "timely" (quicker) initial report.

1. Files
2. Accounts
3. Logs
4. Network Connections
5. Scheduled Tasks
6. Users logged on
7. Registry
8. Processes

Table 3. Suggested Artifacts Comparison Chart

Windows Forensic cookbook [9]	Operating System Forensics [10]	First Responders Guide to Incident Handling [4]	CS4684 [11]	Practical Windows Forensics [12]	SANS [13]	Incident Response and Computer Forensics [14]
LNK files	Files	Files	Files	File systems	Files	Open Files and full file system listing
Prefetch files	System Memory	Current System uptime	Tasks Scheduled	Memory	Scheduled tasks	User login history
Event logs	Logs	System and User Profile information	Logs	Event Logs	Logs	Application and Event logs
Recycle bin content	Web Browsing	Web and Email	Network connections	Web Browser and email	Web browser, applications, and email	Network connections
	Executable programs	Open Connections and Ports, Routing information	Users logged on	Routing table, Arp cache, kernel statistics, network connections	Network Connections	Routing table, Arp table, and DNS cache
	Malware	Users Logged on	Registries	Remote logging and monitoring data	User data, Account Usage	Tasks Scheduled

Windows Forensic cookbook [9]	Operating System Forensics [10]	First Responders Guide to Incident Handling [4]	CS4684 [11]	Practical Windows Forensics [12]	SANS [13]	Incident Response and Computer Forensics [14]
	System Configuration	DLL and Shared libraries	Processes	Registers and cache CPU	Registry Keys	Registry Currently loaded drivers or modules
		Running Processes	Accounts	Process table	Processes (Running and recently used)	Running Processes
				Physical configuration	Physical configuration	System configuration, Startup configuration files, and User profile information

II. VIRTUAL ENVIRONMENT

The constructed Training and Challenge Modules or Labs employ a device particularly well-suited for the job: virtual machines or VMs. One investigates artifacts on a computer running an operating system (OS) of some kind. A VM runs the same operating system code as a physical device and therefore delivers the same interactive experience as those physical machines. This VM can then be altered, saved, tested, and transmitted (i.e., shared). A VM thus facilitates multiple users being able to benefit from the learning environment and scenario that is captured by it. Indeed, this ability to have the essence of a computer in digital form set for use in a training laboratory is so useful, it is widely practiced.

A. SETUP

We chose to construct our VMs in the Naval Postgraduate School Cyber Battle Laboratory (CYBL). This lab's collection of servers is remotely accessible and provides several templates from which to build multiple VMs based on multiple operating systems, as needed/desired. CYBL has several networks; some connect to the Internet, while others are isolated. Isolation serves to prevent any "adversarial" experiments (e.g., malware) from causing real damage by escaping the confines of the experimental environment.

Among the VM choices are Kali Linux (Rolling Box) and several Windows versions with few patches, often having only the 1st Service Pack. We selected a WinOS VM for our Training and Challenge Labs, as this operating system is the most prevalent in United States Navy systems. We chose the Kali operating system for its pre-loaded Metasploit Framework (MSF). This toolset contains everything needed for us to step into the attacker's role of exploiting the Windows VM. In so doing, we created all of the operating system artifacts we desired for our Training and Challenge Labs while acting as an attacker, much the same way as a real attacker creates actual artifacts on a compromised system.

B. EXPLOITATION

1. Training Lab

The exploitation conducted for our Training Lab began by using a highly reliable vulnerability ms08_067_netapi [7]. This top choice among Windows exploits corrupts the stack and enables remote code execution. Although it is not new, many systems remain vulnerable to it. This vulnerability exists within a Simple Mail transfer program by Seattle Labs operating over port 139 and 445. Port 139 is NBT over IP, while port 445 is for SMB (Server Message Block) over IP [7]. The capability granted to SMB and NetBIOS running over TCP/IP is undeniably powerful: one can obtain any piece of system data, no matter how well one may have hidden it. One can also write to the hard disk in a hidden manner. The intruder becomes indistinguishable from a legitimate user, apart from leaving artifacts indicative of malicious behavior. More information about this exploit can be found at the website provided in [15].

The command '**service postgre start**', along with '**msfdb init**' is necessary prior to running MSF. A database must be initialized and ready for use by MSF. These tasks are included in the MSF icon located on the sidebar of the Kali desktop, which is a time-saving step.

The MSF command '**use**' selects the desired module. In this case, the command used was: '**use exploit/windows/smb/ms08_067_netapi**.' SMB stands for the aforementioned Server Message Block.

Actual offensive cyber operators will not clairvoyantly 'know' all the vulnerabilities for systems they target (or defend in the case of defensive cyber operators). Rather, they must perform vulnerability scans to discover if known vulnerabilities exist (whether they aim to exploit them, as offender, or to patch them, as defender). We already knew from course CY4710: Adversarial Cyberspace Operations that the netapi module should succeed against our target VM. Our target was loaded with a SLsmtp.exe from 2003 that allowed the powerful SMB exploit.

Since the goal was to execute our own code, we also had to specify that to MSF by loading a payload. We choose a Meterpreter designed for Windows. The command

used was: ‘**set payload windows/meterpreter/reverse_tcp**’. This powerful tool is designed to “land” (i.e., capture the CPU’s instruction pointer) within an existing process, which can then facilitate any one of a multitude of post-exploitation actions. It is also necessary to input required options in MSF that identify our target IP address and port before launching the exploit [7].

The command: ‘**show options**’ reveals all possible parameters for the chosen tool, and whether they are required or optional. Always required was the origin or Local Host, and the destination or Remote Host addresses. The commands used were: ‘**set LHOST 169.255.177.177**’ and ‘**set RHOST 169.255.197.17**’ [7]. Of course, there were many other addresses used as this value needed to be set correctly whenever the network or target changed. We never employed an option to change the port, as the default of 445 was ideal, should one find that port open; we ensured this was the case for our lab design.

We desired the Training Lab to contain all categories of artifacts with an emphasis on overtness, in order to ensure numerous attack artifacts for discovery by the student learner. For example, we deliberately migrated Meterpreter to a non-existent process in order to create a process artifact. The command used was: ‘**run post/windows/manage/migrate**.’ The more attacker-savvy action would have been to migrate it into a stable process and remain unnoticed, but that would make it a more difficult/obscure artifact for the student learner to observe.

We created a file that had an iconically suspicious name, badRAT.txt. We created a new user account named pwnedU, left this account logged-in, and elevated the account’s access by adding it to the Administrator group. The commands for that were: ‘**net user pwnedU /add**’ and ‘**net localgroup Administrators pwnedU /add**.’ We scheduled outlandish tasks (LaunchTrojan.job), made blatant registry changes, left our network connections open, and left the application and system logs intact while deleting the security logs.

Any attacker who desires to remain anonymous will put forth an effort not to leave behind so many clear indications of a compromise. However, realism in the scenario is traded for quantity of artifacts. Also, the training nature of this exercise calls

for some helpful markers on the artifacts. Because the environment is a virtual machine, it has all the components of a computer. Wading through the mass of data contained in, and manipulated by, a computer with the goal of identifying the suspicious data is a daunting task to the inexperienced. The flamboyant labels of a few select artifacts become conspicuously muted when tucked among tens or even hundreds of thousands of other benign 'artifacts.' This, in essence, becomes a search for a relatively small "signal" (attack-related artifacts) among a large backdrop of "noise" (non-attack-related artifacts).

2. Challenge Lab

The goal of the Challenge Lab is to have the student demonstrate proficiency as an Incident Responder under more realistic, less guided conditions. We consider that the discovery of even a few artifacts that lead to the verification of a CIRCE, along with accurate and informative entries in an incident report, represent a significant demonstration of proficiency lying at the core of what an incident responder is expected to be capable of doing.

Accordingly, our Training Lab exploitation focused on process creation, leaving behind files tainted with malware, evidence of tampering with the Registry in highly suspect fashion, strange users with administrative privileges left logged-on, logs of attempted network connections, and system logs full of events representing our activities. These are needless breeches of covertness for a thorough compromise via a remotely running Meterpreter.

The Challenge Lab, on the other hand, employed a stealthier demeanor to gain access to the system, to avoid purposely creating excessive artifacts. Armed with powerful Meterpreter access via netapi, the simulated intruder-injected malware is set to execute upon user (the student) activity. That malicious execution both creates artifacts, and serves as the abnormal system behavior cueing likely to be noticed by the system's owner. This in-turn would result in the matter being brought to the attention of a first responder. As before, the student is expected to step into that role, investigate the artifacts, and report on the incident with minimal help. The skills learned and tools used

during the Training Lab should be sufficient for the student to be able to accomplish the Challenge Lab with no additional guidance.

The setup for the Challenge Lab consisted of three stages: Pre-tasks, Stage 1, and Stage 2. Pre-tasks consisted of preparing the environment so that the planned exploitation in Stage 1 would occur in a continuous block with as few restarts as possible. Avoiding multiple exploitation sequences takes up less time, and creates artifacts that would be most likely/typical of an attacker who executed one intrusion. This was desired, vice a scenario consisting of a series of penetrations stretching over months of time, as was the case for the Training Lab. The result of a completed Stage 1, then, is the exploited system "template" to be used by the students. It represents a machine infected by malware that has yet to launch. Actions taken by the student will actually trigger the malware, and result in the various artifacts of interest (i.e., related to the attack) being generated. Finally, the purpose of Stage 2 was to test Stage 1 and allowed for alterations to ensure the Challenge Lab performed according to design.

The first Pre-tasks step in preparing the Challenge Lab was to create another Windows XP virtual machine from the same template as the Training Lab. Next, a set of accounts was created using State names to follow the theme set by 'Georgia.' These included Arizona, Connecticut, Idaho, Mississippi, Nebraska, and Wisconsin. The intent was to create a more populated setting resembling a corporation. Here, one finds differing levels of permissions based on group membership. The next Pre-tasks step was to create groups and assign members to them, such as the standard administrators and users. Another group, the Executives, comprised the following user accounts: Nebraska and Wisconsin. The Executives have privileged access to the most valuable company property.

This high-value property is the notional target of the Challenge Lab's imaginary attacker. The next Pre-task step was to 'create' the 'valuable items.' A folder called 'Experiment Results' was made to hold various important files that only the 'Executives' are authorized to access.

The storyline for the Challenge Lab reveals that the attacker was not able to exploit any executive directly, as they were all highly trained and security-conscious to present a soft target for the attacker. However, the attacker was able to access the rest of the system and scheme a plan. An executive would unwittingly copy the secrets to a location open to the attacker. Malicious scripts set in place by the attacker accomplished the information copying and exfiltration. The next Pre-tasks step was to build a repository called 'Deleted Files' that would look innocuous. This would hold the company secrets until they can be exfiltrated to a fictitious IP address controlled by the attacker.

The next Pre-tasks step was script writing. The design for a multi-effect script was to accomplish the attacker's goal. An 'Executive' logon action was set to trigger the script launch. This granted the script permission to access the Experiment results. The script next searched for content and copied it. The script then wrote into "Deleted Files" any content not already existing in that repository. Then the script attempted to exfiltrate the new repository contents to 196.254.33.45. This represented a location accessible to the attacker. RFC 3927 reserves the address range containing 196.254.33.45 for Link-Local addressing [16]. Because it is not allowed to route a Link-Local address to the Internet, the attacker may only be using the 196.254.0.0/16 address as an intermediate step for subsequent exfiltration over the Internet [16]. Since the attacker already has *some* control over this other device in the local network, perhaps he/she also possesses direct access. Further investigation is required to determine what information, if any, left the network, when it left, and where it went. However, this type of in-depth analysis is beyond the scope of the Phase 2 analysis modeled in these labs. The final portion of the script deleted the account previously used by the attacker (Massachussets [sic]) at user log-off.

The scripts utilize wscript.exe to run and this allows them to blend in with the Windows environment. Wscript is a native windows service whose purpose is to run VBScript files. When any script is run utilizing this Windows-based script host it will always appear in the process list or windows task manager as wscript.exe with no amplifying information on the script being run [17]. Therefore, the script being deployed by the service could actually be something malicious, such as a Trojan. This allows the

scripts to hide in plain sight among the scripts that are used for legitimate task, such as system administrator scripts for automating account set-up or printer sharing. The Challenge Lab imaginary attacker has done likewise. From a student evaluation perspective, it is considered acceptable that students simply notice this file as potentially suspicious, without them necessarily discovering the malicious scripts running from it.

Other Pre-tasks steps enabled the smooth development of the virtual machine in a general sense, but did not relate specifically to the storyline. One of them was to enable the Windows logs, unlike the Training Lab where most logs were disabled. This was done through the **Control Panel** under **Performance & Maintenance.** There, **Admin Tools**, then **Local Security Policies** were selected. A window opened to allow changes to **Policy** and **Audit** settings of the WinOS Security logs. The following Security log audit functions were enabled based on what information was desired in the logs as well as best practices provided by Microsoft:

1. Every Success or Failure for: Account Logon Events, Account Management, Directory Service Access, Logon Events, Object Access, Policy Change and Process Tracking
2. Successful System Events
3. Failed Privilege Use and Directory Service Access. [18]

Another needed Pre-tasks step was to configure folder sharing between virtual machines. Only the Windows 7 VM had Internet access, a requirement in order to obtain the SysInternals Suite, as well as FPort and other tools provided for the Training Lab. These tools were then transferred to both the Challenge and Training Lab VMs via the aforementioned shared folder.

Stage 1 was an enactment of the steps taken by the attacker that would result in the creation of the artifacts for later investigation by the student. To begin, the Metasploit framework to make three failed attempts to logon as Georgia. This simulated password brute forcing, or at least guessing, and started populating the logs with the first indications of something possibly wrong. A 4th attempt to login into Georgia succeeded.

Next, the attacker now impersonated Georgia and created a user-level account named Massachussets [sic]. The purposeful misspelling was to evoke that this is a

questionable account, but only subtly so. Then, without severing the Georgia MSF Meterpreter intrusion, the attacker logged into the new Massachussets [sic] account.

The attacker then attempted to breach the Executive-only folder holding the company secrets, as well as other protected objects. These all failed. Next, the attacker created the repository folder 'Deleted Files' with success since the users were permitted such action.

Next, the attacker escalated the privileges of Massachussets [sic] to the Executive group level via MSF. Presumably, the attacker could have absconded with company secrets at that point, but the storyline is that the company might generate items that are even more valuable. Getting the latest versions of those new corporate secrets was the top aim of the attacker. To that end, the attacker implanted a prepared script into the Python2.7/Tools/Scripts folder. This location was another instance of hiding in plain sight. The attacker then set the script's permission to execute. Finally, the attacker utilized a scheduled task to launch the implanted script upon logon of any Executive. The Challenge Lab instruction sheet guides the student to log-in as one of the Executive account to ensure the script launches.

At this point in the VM development, a snapshot was made of the WinXP VM. This was to be the Challenge Lab deliverable: that point where the VM represented an infected system primed for anticipated discovery/detection and investigation action by the company employees (i.e., the student playing that role). All students fulfill the role of Georgia arriving at the next workday.

Stage 2 comprised performing the entire Challenge Lab. Then we refined the scripts until the XP VM performed as expected. The snapshot was then ready for conversion into a template for students to replicate.

C. TRANSFER TO PORTABLE FORMAT

A helpful trait of VMs is that they are transferred the same as any other multi-Gigabyte-sized file. We extracted the Training Lab VM onto our external hard drives directly from the CYBL with the assistance of the technician maintaining it. By employing an .ova or .ovf format, these files can be opened anywhere with the use of software such as VMWare or VirtualBox. Due to their large size, direct downloading with modest-bandwidth available can be difficult. In this case, the VM size was nearly three gigabytes. Other options include hosting the file in the cloud or other file-sharing systems like Dropbox, or even using handheld portable media like flashdrives.

For those with access to the CYBL, a more handy method of working with the VM was simply to create a copy of it from a template we provided. At NPS, the desired format is to use the CYBL, an in-house resource, especially when dealing with malware and exploitation or any other potentially malicious digital tradecraft that should be isolated from any operational public networks. Because the CYBL is accessible by web browser, anyone anywhere in the world with a CYBL account and VMware Horizon Client may replicate the Training Lab and Challenge Lab virtual machines templates for training or future research projects.

THIS PAGE INTENTIONALLY LEFT BLANK

III. INVESTIGATION METHODOLOGY FOR ARTIFACTS

While conducting the preliminary analysis, first responders should always be conscious of what their actions are doing to the system. Everyone should utilize a light touch when attempting to collect needed artifacts. Artifacts; i.e., the data collected during the preliminary investigation, can be broken into two categories: volatile and nonvolatile (or persistent) data. The category in which an artifact falls has a large impact on how and when a first responder collects them.

Nonvolatile data encompasses the data that is unaffected when a system disconnects from its power source, whether intended or unintended. This data is generally stored on the system hard drive, or on removable media devices. Though this data is more stable than volatile data, it is critical to protect the integrity of this data throughout the incident handling lifecycle. This integrity is ensured through a well-defined chain-of-custody practice that includes the generation and verification of hashes of the data as it moves through the lifecycle. Among the eight artifacts listed in Section B of Chapter I, five fall into this category: accounts, files, tasks, registry, and logs.

Volatile data generally lives in system memory. Such data is likely to be lost when a machine disconnects from its power source, whether intended or unintended. One must collect this type of data on a live system before it is lost from memory [19]. Though this data is highly sensitive to system shutdown, there may be times when shutting down the system is the best action to prevent further damage to the network. Because this data is always in fluctuation and affected by system shutdown, a first responder must understand the risk of what may be lost in the case that a system shutdown is deemed the most prudent course of action. Among the eight artifacts in scope for the intended investigation level (i.e., Phase 2) of these labs, the following three artifacts fall into this category: running processes, users logged-on, and network connections.

No matter what type of data one collects, setting a plan in place before an incident happens will ensure the best handling of the artifact data. This plan should include creation of a first responder tool kit, written guidance on what to consider an incident,

and how to handle and report each incident. Within the toolkit will be all the various tools needed to conduct proper system artifact analysis. Tools that are resident on the targeted system may have been subjected to adversarial tampering, and thus may not exhibit correct behavior when executed. Users or first responders may use such tools to conduct the initial investigation into suspect behavior on a system. However, once a CIRCE becomes known or suspected, these same tools may be considered unreliable, as their behavior may have been altered in conjunction with the attack.

The following sections give a brief description of the artifacts suggested in Chapter I, explain which tools to utilize for their analysis, and suggest signs of malicious indicators to seek.

A. PROCESSES

1. Artifact Description

A generic process description is, "the series of steps and decisions involved in the way work is completed" [20]. A process generally involves four basic elements: steps or decisions, variability of processing time or flow, timing and interdependence, and assignment of resources [20]. A computer process works in the same framework. In other words, computer processes are a set of instructions that are allocated some amount of central processing unit (CPU) resources [21]. A Windows process will consist of:

1. A unique identifier called a process ID (PID for short)
2. One or more threads of execution
3. A private virtual address space
4. An executable program
5. A list of open handles to various system resources
6. A security access token. [22]

A process can also parent, or spawn, another process. This is an important fact as often times it may be normal to see a particular process with multiple instances. A good example of this is the process named Service Host Process, or svchost.exe. This is "a generic host process name for services that run from dynamic-link libraries. There are

generally multiple instances of this process in order to improve resiliency of a Windows system should one [instance] fail and to [facilitate] better performance overall" [23]. Once first responders understand the basics of a process, they should also ensure they recognize the baseline (i.e., normal) behavior of their system in order for this artifact to be useful. This can include, but is not limited to, understanding what programs are normally in use, what policies users should adhere to when using the system, and normal CPU workload. This information should be available to responders, as it would aid them in identifying those processes that seem abnormal and thus deserving of further scrutiny.

2. Investigation Tools

Often times, an easy start to understanding the processes running on a system can be accomplished through a quick Internet search. The Internet contains a wealth of information on what a process does, as well as which processes are often used by attackers. The Training Lab proposes three different tools for analyzing processes.

The first tool suggested provides the simplest view of a running process. This is the Windows command **tasklist**. This command opens a command window to show a simple list of running processes. A useful argument to include when using this command is the argument **/svc, which** displays any service (svc) associated with the running processes. The second tool suggested is the task manager, which displays CPU utilization as well as running processes. Lastly, the Training Lab recommends the use of the SysInternals Suite tool PROCMON (meaning Process Monitor) to learn more information about any given process of interest. This information includes, but is not limited to, image path, process tree, PID, start and stop time, and owner of the process.

Due to the method of exploitation used in creating the two labs, as described in Chapter II, the actual process used to exploit the system will no longer be running at the time a student is investigating the system. Consequently, the process is not visible when utilizing any of these tools. For the sake of instruction, the screenshot in Figure 3 was taken while the process was active in memory. Doing so provides the process tree that shows the actual rogue process to the student. This was done so as to edify the student

regarding what would have been visible in the case that this was a live, i.e., in-progress, exploit, vice one that has already terminated.

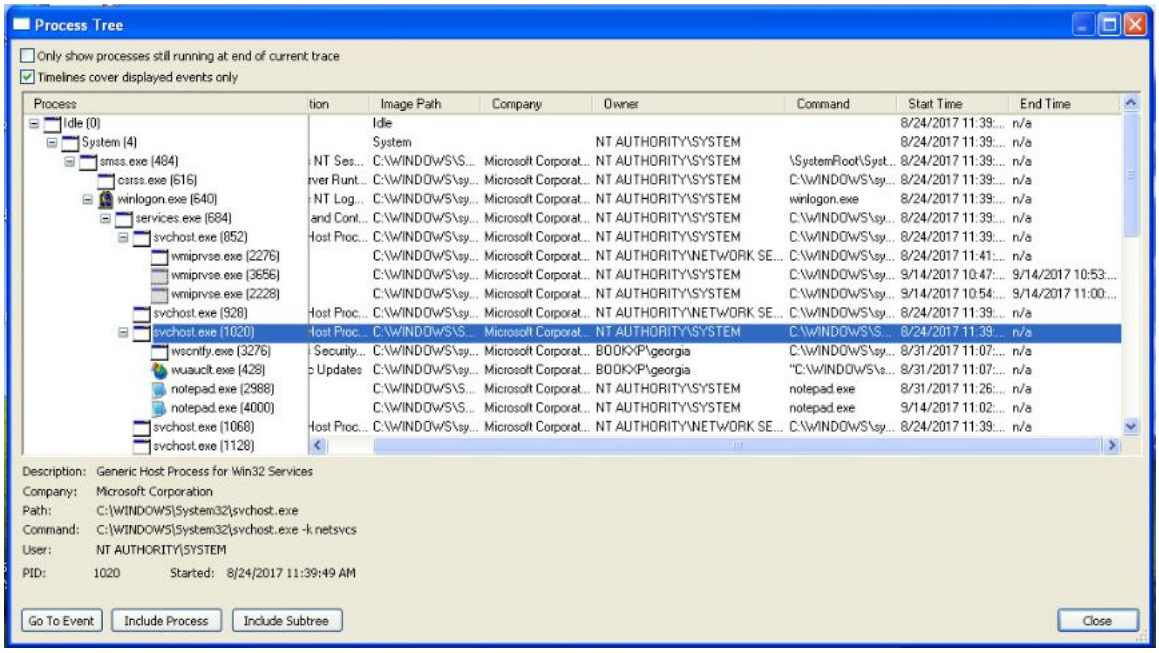


Figure 3. Process Monitor 3 Process Tree View

3. Evaluation of the Indicators to Determine if Artifact Is Malicious

A good indicator that a rogue process, or one that is otherwise exhibiting unusual behavior, is running, is abnormally high resource consumption (i.e., bandwidth, processing or memory). One can view this information with the Windows Task Manger graphical user interface (GUI). First responders should avail themselves with any list the system owner may maintain that enumerates valid and typical processes, or any other such information gained from reviewing organizational guidance, and user interviews. With such information, they can then start to identify candidate processes that could be associated with, or responsible for, the suspicious activity. While utilizing the suggested tools, the student was further instructed to analyze a potentially suspicious process using the below queries listed in [24] to determine the legitimacy of that process.

1. Process' Name
2. Process' Extension
3. Location of Process' Image (File)
4. Process' Parent
5. Process' Children
6. Number of Process Instances
7. Process' User/Owner Account
8. Process' Start and Elapsed Time
9. Process' Command Line Arguments
10. Process' Base Priority
11. Network Connections/Ports Created by Process
12. Interesting Strings found in Process' Image
13. Process' Current Execution Status
14. Process' Hash (as compared to known good reference)
15. Process' Existence When Not Expected (or opposite)
16. Process' Resource Utilization
17. Files Opened and Privileges Thereof
18. DLLs Loaded by Process
19. Process' Persistence [24]

The suspicious process suggested in the Training Lab was an instance of SLsmtp.exe (or Seattle Lab simple mail transfer protocol). A quick Internet search of this process revealed that it is an extension of SLmail, which has multiple remotely exploitable buffer overflow vulnerabilities [25].

The Challenge Lab is configured to launch a malicious script in conjunction with a student's log-in. Though such an artifice mimics what a real attacker may do, the purpose of this script in the lab environment was to ensure that an "interesting" process (to an investigator) runs despite the static nature of the VM that hosts the lab experience.

The Challenge Lab prompts students to identify the suspicious process on their own. The expectation is that the students will determine the legitimacy (or not) of the identified process based upon edifying information acquired from conducting the Training Lab. The scheduled task is set to launch the script every time the student logs in. This ensures the process will not disappear after the student has logged out. This demonstrates behavior of potential persistent malware, and allows the student ample time to analyze the process.

4. Transition Signals

Once a first responder identifies that a process has a high likelihood of being rogue, detecting this process at work on one's system is a strong indicator that a CIRCE has occurred or is occurring. This should prompt first responders to evaluate the totality and bigger picture context of all of the information gathered thus far. If this evaluation reveals enough information to describe adequately the CIRCE, the first responders should terminate the Preliminary Analysis Phase, report their findings, and move into the Preliminary Response Phase actions. These actions may serve to contain or eradicate the rogue process, or—optionally—contain/limit any deleterious effects that may result if the responder decides to let the process continue to run.

B. USERS LOGGED-ON

1. Artifact Description

The 'users logged-on' artifact focuses on identifying malicious users currently logged on the system. Skilled attackers will attempt to limit the amount of time they spend on a system in order to decrease the likelihood of detection. This would also entail limiting the amount of time they must be logged-on to the system. Nonetheless, the investigator would be remiss to not consider that the attacker may still be logged-on to the system being investigated. Consequently, they should check for this.

It is best to use this artifact in conjunction with information discovered through investigation of other artifacts, such as logs and accounts. For instance, the first responder may discover in the security logs that an account was created or modified within the timeframe of the suspected CIRCE. If a user is currently logged on to this account, it

would be prudent to note which files this account is currently accessing or which processes this account attempts to run.

2. Investigation Tools

There are a few simple ways to view this artifact. For instance, **query** is a command line utility used to display the users currently logged-on locally to the system. The Training Lab leads students to utilize **psloggedon.exe**. This executable is a part of the Windows SysInternals Suite. This will not only display locally logged-on users, but also the remotely logged-on users, provided they are logged-on when the tool is run [26].

3. Evaluation of the Indicators to Determine if Artifact Is Malicious

Analysis of users who are currently logged-on will tell first responders which users are potentially related to, or may be affected by, the suspected CIRCE. This artifact may also help establish a starting point for personnel interviews in order to narrow the scope of the incident. These interviews will help establish a collection of authorized user activity that can later serve to eliminate legitimate processes and valid user accounts from the preliminary investigation of logged-on users [27]. Upon identifying a suspicious account, if its user is currently logged-in, the first responder should quickly follow this lead, as this artifact is quite volatile. First responders should attempt to ascertain whether the suspicious account is locally or remotely connected, and what processes or services are currently in use by that account. In the Training Lab, the suspicious (i.e., attacker created) account is viewable from the login screen but will not appear when the student utilizes either of the suggested tools. This portion of the Training Lab demonstrates to the student that not every suggested artifact always bears fruit. In these cases, the student is reminded to not get stuck trying to make something appear. Instead, they should move on to more viable artifacts.

After a student's initial login to the Challenge Lab, the suspicious account will be listed as logged-on. However, once the implanted script runs, this account will be deleted and thus no longer viewable. Since there is no prescribed order to investigating the suggested artifacts, it is expected that some students may see this account, while others will not. This demonstrates the volatility of this artifact. In turn, this may lead to

discussion during classroom debriefing on why each student may have chosen a particular order when analyzing artifacts, and if that order helped or hindered their investigation.

4. Transition Signals

This artifact alone will likely not be sufficient to discern definitively that a CIRCE has occurred. This is certainly true in the case where the attacker is not logged-on at the time the investigator looks at this artifact. It is also less useful—taken on its own—when the users that *are* currently logged-on are known and trusted users. However, when coupled with logs, accounts, and process data, this information may provide enough evidence for transition into the reporting and Preliminary Response Action Phase of the incident handling lifecycle.

C. SCHEDULED TASKS

1. Artifact Description

These tasks are instructions to the OS to run a chosen program in conjunction with some specified event. The attacker may have set up malicious activity that uses some event as a trigger. Scheduled tasks represent one overt way of doing this. One may consider scheduled tasks to be rather overt; i.e., sloppy attacker trade-craft, due to how readily they are located, as shown in the following tools section. However, the careful investigator should not omit checking for such evidence under the assumption that all attackers are skilled, or that—regardless of skill-level—they will never utilize a "blunt instrument" in pursuit of their goals.

Time or logic bombs carry out functions similar to this category, but are much more covert artifacts. Despite the resemblance, they are not task-type artifacts. They nonetheless are worthy of mention as more sophisticated alternatives (to scheduled tasks) that first responders may encounter when dealing with correspondingly more advanced adversaries.

Any task created by adversaries may indicate one of their goals for exploiting the system. The purpose for using a task is likely to launch malicious activity in combination

with some later, anticipated event [7]. A logical reason for such a one-two punch exploit is that the attacker deems these conditions optimal for his or her tailored attack.

Furthermore, an attacker may seek different conditions for various phases of a complex attack. Optimal conditions for exfiltration may be sub-optimal (or worse) for intrusion or exploitation. There may be a security reason why an implant can be put in place most stealthily during a period of (relative) low-alertness. However, the attacker's desired outcome may only be possible when a certain target group is active, which only happens after a maximum defensive condition is in place. An attacker may also want the desired attack timeframe to be well into the future so as to ensure he/she will be well clear of the ensuing aftermath.

When scheduling Windows tasks, one may select a specific timeframe or a specific event to trigger the task. Triggering options include: during a startup, going idle, a workstation lock or unlock, a user session connection or disconnection, when some other task is created or modified, or a myriad of other system event types. A surface level look into the Windows 10 scheduling options reveals that there are over 360 different logs from which to choose a source and event ID in specifying exactly the desired event. A number of conditions and settings can also be applied to further tailor triggering conditions. The available complexity of the task-triggering prerequisites approaches that of a full-fledged logic-bomb.

2. Investigating Tools

Though there is always the option to use a native application, such as the Windows OS Task Scheduler, first responders would do well to employ a 'light-touch' to interact with a potentially compromised system. The SysInternals tool **autoruns.exe** is proposed for utilization during scheduled task analysis. Once this tool is opened, students are directed to click on the scheduled task tab to view any currently running or future-scheduled tasks. A mediocre alternative is to gather this information from the command line command **schtasks**. This would avoid any problems caused by a corrupted Task Scheduler, but not by a corrupted cmd.exe.

3. Evaluation of the Indicators to Determine if Artifact Is Malicious

The task aspects of interest to first responders are rudimentary. Information the responder is likely to uncover can include: the action the task carries out, the author, the time of creation and any necessary privileges or requirements. The two labs present a simplistic environment for evaluating tasks, as there were only a handful of tasks configured into the lab VM. Among these, it is intended in the Training Lab that a first responder will recognize a task called LaunchTrojan.job as malicious. The same holds true for any task that runs a script that violates company policy. In fact, the actions these tasks specify serve as good indicators of the attacker's overall goal/objective for that particular intrusion. Learning a task's author or time of creation are no longer investigative priorities once the existence of such malicious intent (i.e., the explicit or implicit implication of a task's action) has been discovered on a system.

The situation magnifies when the number of tasks increase to levels one might encounter in a large enterprise system. There could be hundreds or thousands of tasks. Searching them singly would probably be futile, especially when there may be only one, or a few, malicious task(s) littered among the many legitimate tasks. Searching through logs is similar, and calls for narrowing the candidates by filters. One way to start is to arrange tasks by date and see which possess creation timestamps within a certain working attacker time-line, or time interval, that has been established from whatever suspicious activity has been gleaned so far in the investigation. A decision is required regarding whether to look at tasks dated earlier than the time of suspect activity detection, or to look at tasks dated later than that detection. This is driven by any sense the investigator may have regrading whether it was a task that caused the suspect activity, or whether an attacker may have created a task post-intrusion, perhaps to cause follow-on damage or to maintain access. If detection occurred very recently, one could simply work backwards from the present. The task list needs no further reduction when the investigator can scan every action or author therein in a few minutes. **Autoruns.exe** is an indispensable tool to help accomplish this feat, as it is capable of sorting task by date.

In general, first responders should consider any task designed to start an activity that violates policy to be malicious. Activities matching this criterion would be those that

are unauthorized, criminal, capable of causing damage, or known as authored by an adversary. This latter example—authored by adversary—would be suspect for any task whose creation time coincides with that in which the adversary was logged on; or when all authorized users have been interviewed so as to rule out any task they made; thus leaving only suspect tasks. Such recognition will depend on the skill and experience of the first responders. They must pull together clues from other artifacts and work to obtain a strong understanding of what is and is not normal on the system under investigation.

4. Transition Signals

A task that is palpably malicious is clearly a mandate to notify appropriate personnel, and to either draft, or assist in the drafting of, an incident report. This task may be the primary vector or root cause for the exploitation, directly achieving the attacker's goal. Alternatively, it could accomplish an adversary's secondary purposes. Examples of secondary purposes include helping the attacker to maintain access, extending the main attack via lateral movement to other systems, or launching diversionary activities. In addition, such discovery will almost certainly happen after a set of other artifacts have been found and used as initial leads that led to the discovery of the task(s).

The two labs created in this research such 'palpably malicious' sorts of tasks. There is little question that the tasks woven into these labs are intended to compromise the system and benefit the attacker: The Training Lab launches a Trojan, and the Challenge Lab runs a malware script to steal intellectual property. The difference between our lab examples and those likely to be encountered in a real-world enterprise scenario, is that the sparse artifact environment of our labs did not provide any cover for these already blatantly offensive tasks. Given this environment, then, a student can readily discover these tasks by searching for them right at the beginning of the lab. However, the storyline purports to lead them to the tasks in a manner that is more typical of a real-world scenario. As one of the non-volatile artifacts that may need some cross-referenced data (clues), it is placed, i.e. brought to the attention of the student, towards the latter part of the Training Lab investigation.

There is prudence in noting that the task activates a Trojan or script that attacks the company secrets. However, trying to get further details into the Trojan or script mechanisms is a delay that the first responder should avoid. If a task was in a grey, questionable status that aroused suspicion, this could be noted for follow-up, but should not hold up the search for further, more conclusive, evidence of an incident. A premature transition could result in a false-positive report, or may lead the first responder to overlook other artifacts of interest that might have otherwise been discovered shortly thereafter.

D. ACCOUNTS

1. Artifact Description

An account is a collection of permissions and settings. An operating system uses it to grant resources to any persona (human or system) able to successfully authenticate for a particular account. Should an adversary have physical access to a system, he/she may be able bypass the account system to gain unauthorized access to it. However, under the more likely scenario, involving remote access to a system, one encounters the operating system. One of the jobs of an OS is to manage the allocation of system resources and objects (e.g., directories and files) to the requesting subjects in a manner that adheres to an access control policy. Accounts are the basic mechanism by which an OS can assess whether such requests are granted or not. As such, accounts play a pivotal role in the overall security posture of any automated systems.

Subverting the access controls facilitated by accounts is the desired outcome for several malicious feats: privilege escalation, credential theft, rogue account creation, or even the obtainment of direct physical access to the system, if/when such direct access averts the need for account authorization. If an attacker cannot obtain a legitimate account credential, or otherwise subvert the OS to bypass its access control mechanism, he/she will not obtain logical access to the targeted system objects. Lack of such access renders subsequent attack objectives (e.g., file theft/exfiltration) attacks infeasible. An attacker must solve this security obstacle early on.

The artifact aspect of an account is rather straightforward: is the account itself, or one or more of its permissions/privileges fraudulent? Answering these questions is the essence of the analysis of this artifact. The system actions (i.e., processes executed or attempted executions) initiated by an account is the primary way to determine if an adversary used it. Besides specific malicious actions taken under a particular account, the investigator should also look for the simple existence of accounts—particularly admin-level or higher—that are not recognized as legitimate by the system's authorized owner/operator.

2. Investigation Tools

A single tool is proposed for analysis of WinOS accounts: **net [user]**. There is a family of commands under **net** that can disclose group membership. There are also several management operations. The intrinsic characteristics of accounts are simple enough for one tool to reveal them:

1. The name of the account.
2. Who created the account?
3. When was the account created?
4. What privileges are granted to the account?
5. To which group(s) does the account belong?
6. When was the account added to those groups?
7. When was the account deleted, and by whom?

The onus is on the investigator to put the account into context. None of the answers to these questions provide automatic proof of something malicious in and of themselves.

3. Evaluation of the Indicators to Determine if Artifact Is Malicious

Despite the possibility that an account may appear completely legit, an account may be suspicious simply by existing. Accounts pertaining to; former employees, deceased or fictitious persons, corporate competitors, a known bad actor, or anything else that is otherwise nonsensical, are highly suspicious. Also suspicious is the creating of

accounts outside of normal work hours. Any account which gained access to an otherwise "closed" or privileged group, such as an eighth member of a strictly seven-member board, is a likely a red-flag indicator of foul play.

The Training Lab account pwnedU is an example of a blatantly suspicious account. The purpose of that account was to pretext the detection at the beginning of the Training Lab. This triggers students to begin their initial first responder actions.

Such account artifacts could be attempts at self-concealment, as when using an impossible identity. An account associate with a nonexistent employee or a person who is deceased are examples of an impossible identity. Whatever the motivation or cause, these artifacts serve the first responder by presenting indicators of a security breach. Pursuing the genuine identity of a false account is a matter for data forensics.

The Challenge Lab account "Massachussets [sic]" blends in among a slightly more developed environment. Here, there were several legit accounts, which all have U.S. state names. Some of the accounts have a special group membership called "Executive." After having learned this group had access to the targeted company secrets, the adversary created an imposter account with the misspelled name "Massachussets [sic]." This account was elevated to an administrator account, which grants privileged access to modify the system.

4. Transition Signals

The Training Lab pwnedU account was designed to be the detection trigger. Per the storyline, this trigger was utilized to move the student from the role of an employee or helpful sysadmin, to the role of investigator or first responder. Even a rather superficial (surface-level) investigation of this account would show that the account privileges were escalated to administrator. On its own, discovery of pwnedU is sufficient to declare a CIRCE per the CJCSM. If this artifact were created by an adversary, then a first responder stopping upon discovery of the account precludes him or her from any knowledge of what the adversary did besides gain access. This could be the sole item in the CIRCE report. Yet, without knowing what the intruder has done, it is not possible to determine what specific damage or adverse impact to the system has resulted from the

intrusion. Local guidance may prioritize a speedy initial report or it may encourage the responders to continue several more minutes if they expect it will be fruitful.

The Challenge Lab's Massachussets [sic] account; similarly, does not possess any basic attributes that would immediately signal transition. At first glance, it looks like yet another routine user account created by administrator Georgia that has followed the basic naming convention established by the company. However, the fact that it was created so much later than the other accounts (abnormal), so quickly escalated to an administrator (abnormal), and, especially, that it joined the 'Executives' should concern first responders. Additionally, administrator accounts should not just "appear" without the concomitant formality and publicity that one would expect surrounding the creation of such a privileged user. Discovering that Georgia was compromised, and either of the two facts that Massachussets [sic] 1) took company secrets or 2) uploaded a script to automate that process; ought to quell any doubt that Massachussets [sic] is indeed a malicious account-based artifact.

E. REGISTRY

1. Artifact Description

Registries are logically organized collections of attribute-value pairings that support as the multitude of system configurations that an OS requires for proper functioning. Registries provide a treasure trove of potential system-wide artifacts that are of use to an investigator information; but it could also take a person an interminably long time to dig out all such digital evidence from these intricate data structures. This artifact merits investigation with caution as unintentional changes to these values could cause user-level or system-level malfunction or breakdown [17]. For the WinOS, information such as application configuration data, system configuration changes, removable media utilization, wireless connection settings, and passcodes, are all stashed in the Windows Registry. Each can be very helpful to first responders if they know where to look.

The Registry divides into multiple system-specific and user-specific divisions called "hives." Each hive consists of three components: keys, values, and data. The Incident Response and Computer Forensic book compares these components to that of a

file system; where a key could be thought of as the directory path, values as the filename, and data as the file contents [14]. The main five registries of the system are SAM, SECURITY, HARDWARE, SOFTWARE an SYSTEM. The five major hives are located in **%SYSTEMROOT%\system32\config** and two user-specific hives are located in the user's profile directory. When the system is running, Windows maps the contents of the hives into a tree structure that begins with a set of root keys. In turn, these further divide into subkeys, which represent the hive files. These root keys are a good place for first responders to begin investigating the Registry [14]. The following list describes what information is stored in each root registry key:

1. HKEY_CLASSES_ROOT (HKCR)—This hive stores information about drag-and-drop rules, program shortcuts, the user interface, and related items.

2. HKEY_CURRENT_USER (HKCU)—This hive is very important to any forensic investigation. It stores information about the currently logged-on user, including desktop settings, user folders, and so forth.

3. HKEY_LOCAL_MACHINE (HKLM)—This hive can also be important to a forensic investigation. It contains those settings common to the entire machine, regardless of the individual user.

4. HKEY_USERS (HKU)—This hive is likely to be critical to forensic investigations. It has profiles for all the users, including their settings.

5. HKEY_CURRENT_CONFIG (HCU)—This hive contains the current system configuration. This might also prove useful in your forensic examinations. [14]

2. Investigation Tools

A listing of Registry data can be obtained from a command prompt window with the **reg** query command. Since not all Registry data is human-readable, two SysInternals Suite investigation tools are suggested to the student. The first is **autoruns**, which displays the desired registry keys data value if the value is set. From there, students can navigate to the Windows Registry editor, **regedit.exe**. This program allows students to view a desired key's location in the registry along with any information that has been stored in the various hive components. Simply starting in the Registry editor may not always be the best course of action as there can be an overwhelming amount of

information to navigate through. A more surgical approach of pairing these two tools will help to focus the first responders' Registry investigation, in addition to saving time.

3. Evaluation of the Indicators to Determine if Artifact Is Malicious

Unexpected key values or data in the registry, or changes to known values, are strong indicators that system tampering may have occurred. Changes to the registry can result in system crashes or abnormal behavior, which are also indicators of suspicious activity. In the Training Lab, there are no intentional changes in the registry. There are; however, indications of the creation of the suspect user's account and its permission elevation. There is no information about the attacker's actions available in the registry that has not already been identified by other artifacts.

The Training Lab focuses on developing students' basic understanding of how the WinOS Registry is structured and on presenting some easy-to-follow guidelines on how to conduct a surface-level analysis of them. Students leverage these skills and information during the Challenge Lab. In order to ensure students are ready to inspect the Registry in the Challenge Lab, students perform a guided examination of the registry in the Training Lab.

The **req** query command lists the details of each of the five hives. **Autoruns** and **regeditor** allow for investigation of system registry keys that were utilized to run a task; i.e., the Run, RunOnce, and AutoRuns keys. These could provide useful information to ascertain the attacker's objective or behaviors. The Challenge Lab utilizes the keys mentioned above to ascertain whether the attacker implemented some manner of persistence for his/her attack script.

4. Transition Signals

The registry is attractive to the attacker for a few reasons. First, the amount of system-generated "noise" (i.e., all of the "normal" events generated) that an investigator must sift through in order to find useful artifacts that are directly linked to an attacker's behavior is substantial. The system is constantly producing registry changes in support of numerous events occurring on a normal basis. This noise allows the attacker to hide

virtually in plain sight. Furthermore, the attacker can take advantage of the registries' unique ability to run a program, without requiring the attacker to implant a file or schedule a task. This allows the attacker to achieve persistence for his/her attack, along with a degree of stealth. Alternatively, the attacker could achieve certain objectives without having to add anything to the Registry; but rather, could simply modify or delete a small portion of the registry to achieve the desired system behavior. For instance, if the attacker's goal was to simply degrade the target's ability to use the system (i.e., degrade/deny service availability) a small change in the Registry could achieve this goal, and may be quite difficult for the system administrator to diagnose and correct.

Unexpected programs, previously identified interesting strings, or unknown executables listed in key values are candidate signals that could prompt the first responder to declare CIRCE detection, and to thus transition to the next phase of the incident lifecycle. Due to the size and complexity of the Registry, along with the typical level of Registry knowledge expected from an individual serving in the first responder role (vice digital forensics analyst), the depth of analysis may be insufficient to justify the declaration of CIRCE detection even though there may be abnormal indicators discovered. First responders should keep in mind that Registry hunts can become a very time-consuming endeavor. If the suggested surface-level analysis does not reveal enough information to understand what is happening, the first responder should annotate the finding and attempt to correlate with another artifact to get to a transition point. Phases 4 and 5 of the incident handling lifecycle will allow for deeper analysis of the registry keys and annotated evidence.

F. FILES

1. Artifact Description

Files are the means to store data for a computer or information system. They are part of the routine landscape in the cyber domain. When an adversary creates on, uploads to, or downloads from, a file on a system, it is done so as to achieve, or help to achieve some malicious objective. A file also becomes an artifact when an adversary leaves upon it any trace of his/her interaction with it. Perhaps they stole information, read sensitive

information, altered a file, or embedded an alternate data stream into it. Investigation of particular files can help first responders build a picture of an attacker's activities. Link files, Prefetch files, and files found in the Recycle Bin are examples of files that reflect recent file-related activities [14]. A brief description of these files is provided below.

'Deleting' files does not mean one immediately removes them from the system. The files' index, or memory address, in the file directory system, is flagged in the system as available space, even though the 'deleted' data still exists at that location. 'Deleted' files are copied to the Recycle Bin. As their former storage locations on the hard drive are overwritten, they become progressively unrecoverable from those original locations. It is possible that an attacker has deleted files that could reveal either his/her direct activities, or be indicative of his/her ultimate objectives on the system. It may be possible to recover these files either from the Recycle Bin or from disk space formerly allocated to the file folder [12].

Link (LNK) Files, "act as pointers to other files or folders on a system and are used to create a direct link to an executable file instead of requiring navigation to the file directory" [14]. The location of these files will differ based on operating system (OS) version. In the Windows XP OS, utilized to build the virtual environment, the link files reside in C:\Users\%USERNAME%\AppData\Roaming\Microsoft\Windows\Recent\ and C:\Users\%USERNAME%\AppData\Roaming\Microsoft\Office\Recent\. The link files provide the following information:

1. Full file path (at the time the link was created)
2. Network share name (if target file originated from such a source)
3. Serial number for the source volume
4. Attributes and logical size
5. Standard Information Modified, Accessed, and Created timestamps for the referenced file at the time it was last opened.
6. A unique object identifier (ObjectID), also stored in the target file's Master File Table (MFT) record and used by the Distributed Link Tracking service. [14]

A prefetch file exists for each executable and is stored at SystemRoot\Prefetch. As described in a previous section, this file design speeds up system performance and will generally contain the most recently utilized Dynamically-linked libraries (DLL). Along with the most recently utilized DLLs, the prefetch file will contain the following information for up to 128 executed programs:

1. The executable's name
2. The path to the executable
3. The number of times that the program ran within the system
4. The last run time
5. A list of DLLs used by the program [12]

Time or logic bombs, previously mentioned in Section C, are another example of a file artifact. In general, a logic bomb is a malicious code/file inserted into the system that checks for a particular condition, or set of conditions, before executing some attacker-defined actions. Time bombs work in much the same way, but rather than waiting on any number of general system conditions (as does a logic bomb), it is set to activate on a particular date and time. Though first responders would definitely want to find/identify any such files, actual analysis of them should wait until the Analysis Phase. When being analyzed, it is good practice to do so in a sandbox environment. This is a precaution to prevent further system damage should the file "detonate" upon some built-in logic that detects tampering or discovery.

2. Investigation Tools

A file integrity checker is a great tool to have in place on your systems. These tools offer a time-efficient way to track changes made to files and file permissions. When no file checker is in place, timestamps of files are another good indicator to check when attempting to identify signs of malicious files, or signs of tampering with legitimate files. The Windows file system is organized in a hierarchical structure that makes it easy to search for and view the files and directories it contains.

The MFT is another feature of the Windows file system, which provides a view of the metadata associated with each file. Metadata stored in the MFT is helpful for quickly

gathering file-identifying information; such as name, timestamps, location, and even data if the file is small enough, i.e., less than 512 bytes. [19]. Utilizing the search function of Windows allows first responders to sort by date or file type, such as .lnk files, and thereby narrow their file search space.

A .pf reading tool, such as WinPerfectview, opens Prefetch files in a human-readable format. One recovers deleted information through examination or restoration of Recycle Bin files (whether viewable or hidden), or with an imager such as AccessData FTK [12].

3. Evaluation of the Indicators to Determine if Artifact Is Malicious

Indications of malicious behavior concerning files can include file creation, file modification, or file deletion. This artifact is more valuable to the investigator when it is considered in the context of written guidance for system usage and user interviews; as these help establish a baseline of "normal" against which abnormalities are more likely to be noticed. First responders should also be suspicious of any unusually named files. In the Training Lab, the suspiciously named file badRAT.txt is hidden in the system32 directory.

In the Training Lab, the student is prompted to leverage information learned about an already identified CIRCE to identify suspicious files through a combination of time-scoping and a simple Windows search of all files. Even though over 30,000 files exist on the system, most of them ought to have been temporarily removed from the investigation platter. The account suspected of creating badRAT.txt was itself created in recent months, but most files on the system have a much older creation date. That file's name and metadata act as signals that the file is potentially malicious, or foreign to the system.

In the Challenge Lab, the attacker did not merely implant a file in the directory, but also attempted to exfiltrate data. The student is expected to again scope the incident based on already gathered evidence, such as time evidence found via the accounts or logs artifacts. From there, students are to trace the attacker's movement through the file system. Metadata and timestamps are the key elements to this analysis.

4. Transition Signals

Depending on what is discovered in the file system, this artifact alone may not be enough to signal transition. For instance, a newly created file may, depending on other contextual factors, warrant further investigation; even though its existence could be benign. Unless an unauthorized, or authorized but hijacked, account created the file, this rather routine file-creation event certainly does not warrant transition. However, the discovery of a known malicious or suspiciously-named executable file would likely warrant transition no matter who authored it, or when that occurred. Given the potential danger that exists with malicious executable files, we expect the student, acting in the capacity of a Phase 1 (Detection) investigator, merely to find/notice suspicious files, not to actually analyze them. Actual analysis is left to more advanced responders (e.g., forensics and/or malware analysts) who have the specialized training and tools needed to do so. This artifact becomes stronger by correlation with other system artifacts.

G. LOGS

1. Artifact Description

Numerous logs record events on a host machine daily. In the case of a Windows OS, the three main, or core, logs are Application, Security and System. Each is a great tool for rebuilding the history of what took place during a suspected incident. The Security log records events that reveal attempts to enter the system, such as login attempts, and logouts. The Application log records events relating to specific application processes running on the system. The System log records information that pertain to system-wide events, such as kernel operations, power management, disk drives, time, and other services that always run as part of the basic “infrastructure” of the system. All of these are recorded and ranked on a three-tiered priority scale. On this scale, the lowest priority is information, the middle priority is warning, and errors are at the top of the priority scale.

Based on how the logs are configured in a system and the volume of events generated, there could be hundreds of thousands of logged events to sift through. Utilizing information and leads gleaned from the inspection and consideration of other

incident-related artifacts, one can significantly narrow the log search space to only what is likely to be related to the suspect CIRCE. Indeed, this tactic is suggested to the students. By focusing their investigation of the logs into a particular time period, an otherwise intimidating quantity of log data begins to narrow to a manageable level. Unusual events can also be great indicators of where one should look next. A blank log period could imply deletion of existing logs, or disabling of logging. In either case, these are fairly reliable indicators of malicious activity. Any such unexpected changes in the status of system logs can help an investigator establish a timeframe for when the attack occurred. In their absence, these artifacts could signify an attacker's desire to hide his/her activity on the system.

2. Investigation Tools

The first tool suggested for searching logs is Windows Event Viewer, accessed by typing **eventvwr.msc** into a command prompt window. This is a standard component of the Windows OS. It opens in a pop-up window and presents a graphical user interface consistent with the Windows environment. It provides access to the three core Windows logs already mentioned, along with two specialty logs named 'setup' and 'forwarded events.' Logs created by any add-on (i.e., not a native part of the OS) programs installed on the system may be observed with the Windows Event Viewer. There is also the 'administrative event' custom view, which shows a superset of the core logs by combining events from any source log. Naval Postgraduate School users can also utilize the Windows Event Viewer to view logs created by activity association with the virtual private network used to tunnel onto the school's intranet.

Despite the high quantity of events logged, Event Viewer is a tool with which all beginning incident handlers should become familiar. As the Windows Event Viewer exists on every Windows system, it is reasonable to assume it will always be available to view desired log entries. However, the fact that it is a native windows program does not mean the viewer is incapable of being corrupted. This tool is simply suggested as a likely starting point when attempting to find event "needles" of investigative utility from among the "haystack" of all the system events recorded in the logs.

The Training Lab next suggests the log tool **PSloglist**, a SysInternals Suite command-line tool. It allows one to view log data in a human-readable output, or to dump logs into a plaintext (Excel) file for a searchable review capability. This tool carries the advantage of being useable in a read-only, off-system manner, as part of a forensic workbench, or more simply a CD-ROM. When employed in such a 'light-touch' manner, **PSloglist** reads the target system without changing it. Thus, any forensic image of the target machine obtained after viewing its logs using **PSloglist** remains faithful to the state of the machine prior to such viewing.

3. Evaluation of the Indicators to Determine if Artifact Is Malicious

Several broad indicators are suggested during the lab, including failed log-on attempts, startup or termination of services, remote connections, created or deleted accounts, startup or termination of processes. As previously noted, when expected log entries are not present, it is likely the result of a human act. The events themselves may have been suppressed, the logs may have been altered, or the audit policy may have been changed. The ability to detect log aberrations requires a thorough sense of what 'normal' looks like in a system, or at least that part of the system covered by a particular log.

Developing such a keen sense is a product of more time and experience than a laboratory setting can impart. The labs acknowledge this aspect of how best to use logs. However, there was no deliberate altering of logs, whether pinpoint changes or entire log deletions.

The Training Lab VM has very few logs enabled. Nevertheless, there are a few interesting log-based artifacts for the students to view which correlate to other types of artifacts found in the lab. The creation of the pwnedU account was discoverable on the date 17AUG17. The creation of badRAT.txt was also recorded, on the earlier date of 10AUG17. Although the existence of both artifacts should have been previously noted by the students, it is (likely) only after log artifact review that they learn that the suspected Trojan predates the pwnedU account by one week. This is a clue that adversarial activity began before making pwnedU an administrator. The application log events also indicate

the times that the remote connection manager started and stopped, and these times coincide with the attacker's activity on the system.

The other notable log artifacts are indicative of a prominent amount of activity by one program beyond the all others: SLadmin. If this program is unknown, an investigator can easily research it, or request such from more advanced forensic analysts. If researched sufficiently, one would find that its flaws were exploited by malware. However, this exploitation was done as a part of instrumenting the lab so that the lab exhibits the sort of artifacts expected of an attacked system. The student is not expected to conduct the more advanced malware analysis that would reveal the details of this exploitation.

The Training Lab demonstrates that all these events were already introduced and a CIRCE should have already been declared. Accordingly, the storyline uses them as amplifying information suitable for following up with a simulated initial incident report.

A change was made in the Challenge Lab VM's audit policy to enable the security logs. The goal was to provide an abundant amount of information for the students to narrow down via well-chosen filters. Unlike the Training Lab, the exploitation in the Challenge Lab follows a planned and scripted single-episode intrusion for the students to investigate. By narrowing investigative scope down to either the actor 'Massachussets'[sic], or the right time period; it is desired that the students piece together a portion of the attacker's activity. Because the Challenge Lab allows students free-range regarding which artifacts they pursue, and which leads they follow, it is left up to them whether to search the logs earlier in their investigations, or later. If later, they may have already determined what incident occurred and may have only sought corroborating evidence to add detail to their reports.

The following log artifacts are available in the Challenge Lab: failed attempts to logon on as Georgia, the creation of an Executive account much later than the others, failed access attempts to restricted folders, followed by successful attempts of those folders, copying of data from the 'Experiment Results folder, the creation and injection of a malicious script, that fact that launching of that script coincides with the user's logon,

and unsuccessful network connection attempts to exfiltrate data from the attacker's suspected repository. Students who discover these artifacts gain much insight into the attacker's actions.

4. Transition Signals

Should any one of the Training Lab log artifacts mentioned above be found, this could be (subjectively) considered sufficient evidence to make a determination that a CIRCE has occurred. Should a greater number of these log artifacts be found, their combined correlative effect creates a much more convincing transition signal.

By searching the logs earlier in the Challenge Lab investigation, transition signals could come from uncovering several failed authorization attempts against the Georgia account. If the student then also discovers the newly created account, and that account's attempt to access the more sensitive Executive-only objects, then he/she should discern even stronger evidence of CIRCE activity. Certainly, the discovery of script implantation should, by itself, exceed the incident threshold for most investigators.

H. NETWORK CONNECTIONS

1. Artifact Description

Abnormal or unexpected connections could signal unauthorized remote access that requires further investigation. A skilled attacker will attempt to limit the time the connection is in use to prevent his/her own discovery. The combined effort to make a brief connection and to choose the most inconspicuous period to do so (for example, when no one is monitoring the system, or quite the opposite, during peak periods when there are many other connections to hide among), reduces the likelihood that first responders will notice such connections.

This situation changes if network connection artifacts are also logged. Indeed, such logs make it trivial to review the connections post-facto. Nonetheless, this entails a number of requirements: the audit policy must have been on, the logs must not have been altered or deleted, and the attacker must not have convincingly impersonated a trusted connection during an acceptable hour or day (if applicable).

2. Investigation Tools

Netstat is a built-in Windows command-line tool that is highlighted for use in both Labs. It is used to determine if any unexpected, live communication links are established with the local system. If the attacker is active during the investigation, this tool provides the remote connection origination IP address and port number associated with that connection.

As previously stated, having a log of network connections provides a list of historical connections. This could also provide the source IP address and port number used by the attacker. Additionally, there should be a time or date and possibly duration associated with each connection. Logs provide the benefit of preserving volatile events, and remove the need to catch malicious connections serendipitously while active/live.

3. Evaluation of the Indicators to Determine if Artifact Is Malicious

Evaluating an incoming network connection requires information about its source IP address, the port numbers—source and destination--involved, when it took place, how long it lasted, and in some cases, what other system activity may have coincided with it. Should the suspicious connection be outgoing, the destination IP address becomes key.

Should the source or destination address have a history of malicious activity, or be otherwise associated with an adversary, competitor, or other noteworthy entity; then the responder has a clear and strong indicator that the connection is malicious. Recognizing a suspicious address is a difficult task, unless an address is a repeat offender that has not (yet) been blocked. The aid of a database of known bad addresses would be indispensable in helping to decide this.

There is a good likelihood that tracking down the original IP address of the attacker involves multiple steps. The path the suspicious connection took from the attacker's system to the target system will involve multiple hops along multiple routers through the Internet. The attacker can also try to confuse or conceal his/her movement through cyberspace by designating which router his/her traffic will pass through. An attacker can use this behavior to conceal their identity or impersonate another. Other obstacles that help hide the attacker's connection path could include the use of a Virtual

Private Network tunnel, or Onion Routing. The amount of time and work involved in identifying such path information is excessive for the investigator in a first responder role. Instead, this type of sleuthing work is more fitting for Phase 4 of the incident handling lifecycle.

Port numbers have much more utility for the first responder than an unknown IP address. Resources abound that track malicious use of certain port numbers. All ports associated with numbers under 1024 are the so-called 'well known' ports. These ports are widely accepted as being associated with particular services. Knowing that a well-known port was used may instantly reveal the service to which the connection was intended. That is precisely the service that may have malicious code or a vulnerability for the attacker to exploit. Unlike the well-known ports, it is less obvious (clear) what service an attacker targets when using a random ephemeral port to establish the connection. Often, an ephemeral port is used when malware initiated an outbound connection, when opening the port is hardcoded into the malware, or when the port number was somehow sent to the attacker. Unless the port used is well-known or otherwise closely associated with a service, the first responder should pass it to follow-on investigators for future analysis. The VMs used were replete with a variety of ports known to be associated with malware. It is good for the students to notice this, though all open ports are not highlighted in the Labs.

Finally, knowing the start and finish times of a connection helps the first responder in a couple ways. First, if a connection falls into an unusual period, it becomes correspondingly suspicious. Should the system never, or rarely, get remote connections when the business is closed, any such connection would clearly stick out and warrant further investigation. Even if there are no such downtimes, the connection period (start-finish) should receive close attention, as this may be found to overlap with other artifacts associated with attacker activity.

For the Training Lab, the storyline leads the student to suspect that a remote connection was used to create the suspicious admin account, pwnedU. This artifact is highly volatile: a connection must have been established at the time the tool was run in order to be observed by the tool. Due to the volatility of this artifact in our pre-captured

and recorded attack environment, the screenshot in Figure 4 is provided to the student so that he/she may see what the active connection would have looked like when using the netstat tool. The screenshot captures a moment in time coincident with the attacker's network connection.

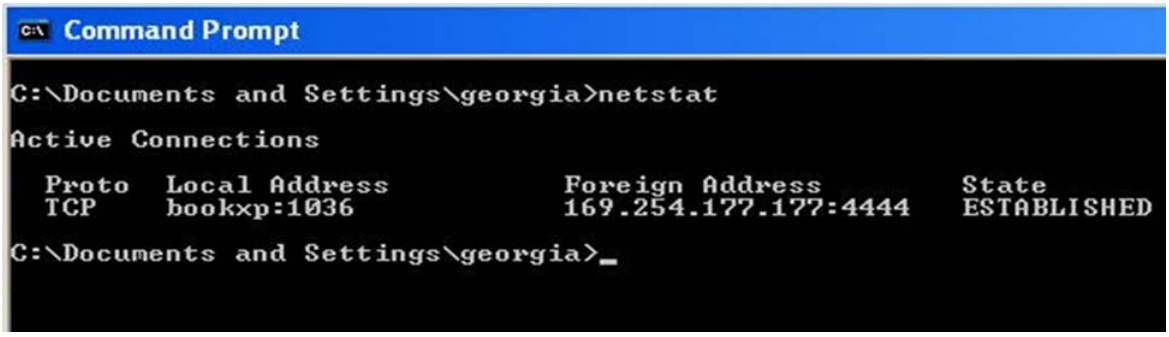


Figure 4. Network Connection Screenshot

By contrast, the Challenge Lab does not overtly reveal the network connection used by the attacker. In a strict sense, this artifact went unused. There is; however, a good amount of connection-specific information in the Challenge lab logs. One of the scripts written for the Challenge lab writes to the logs each time the user logs on. It provides the following message as a system log warning, "A remote connect was successfully made with 196.254.33.45". Even though the Challenge Lab is free-ranging, only a casual search through the logs is needed to discover that there were suspicious network connections made.

4. Transition Signals

A network connection may be the starting point and the best means of unveiling the origin and persona behind an attack. However, the investigation required to determine the persona behind a certain socket pair falls outside the first responder's purview.

Identification of the attacker is not the primary goal of the first responder, rather he/she should attempt to examine and report as many facets of the network connection as possible. This information includes, but is not limited to, IP addresses, ports, and services that are associated with the connection(s). This information can help the first responder

determine information that will help answer questions such as delivery vectors and system weaknesses. Specifics may also be leveraged to construct indicator-of-compromise (IOC) rules that will catch similar attacker behavior in other portions of the enterprise.

A straightforward case may be the result of an attacker who did not have the skill or opportunity to cover up his/her tracks. This could also signify he/she was attempting to divert the first responder from the primary interest/goal of the attacker. Trying to determine which is beyond the scope of a first responder.

Nonetheless, to identify a malicious network connection is one way to prove a remote-access breach exists on the system. That alone signals a CIRCE and encompasses a major portion of the first responder's duty. Prior to transition, the first responder should check if any other artifacts directly relate to the suspect connection. Trying to flush out the main aspects of an attack is suitable for the first responder; attempting to provide every detail is not. Delaying the report only delays the digital forensics team from starting.

IV. CONCLUSION

A. SUMMARY

As the Cyber Domain continues to grow in use and capabilities, well-trained cyber users are critical to maintaining the confidentiality, integrity and availability of systems that are vital to mission success. This is especially true for DOD and other federal government employees who, in the Cyber Age, rely more and more on information technology to get the job done.

The incident handling lifecycle outline in the Chairman Joint Chiefs of Staff Manual 6501.01b provides foundational guidelines on how Department of Defense organizations should systemically approach cyber incidents. Each of the six phases outlined in the incident handling lifecycle help an organization develop and maintain healthy and secure information systems. The training developed in this research concentrated solely on the first and second stages of the lifecycle, which focus on detecting (Phase 1) and identifying (Phase 2) the signs that a Cyber Incident or Reportable Cyber Event, otherwise known as a CIRCE, has occurred or is occurring.

The virtual training environments designed as part of this research could provide additional levels of cyber system awareness to any existing program-sponsored training. These types of virtual environments are unique training tools that can be easily built, installed, and modified to meet the specific training needed. As the training environments are further developed, these VMs can be configured to more closely reflect the system(s) the first responder will be assigned to. This can include, but is not limited to, designing the VM to match the operating system, network layout, antivirus, snort rules, etc. of the system the first responder will be assigned to protect. Furthermore, the training storyline can be easily modified to accommodate new scenarios that reflect nascent, real-world attacker activities, as described on front-page news sources. Each new scenario can provide additional experience for a responder to hone artifact analysis skills. The principles of basic WinOS artifact investigation contained in the two labs serve as an

entry-level debut for what is a long journey toward the development of a highly proficient cyber first responder or dedicated digital forensics analyst.

The created virtual training environments aim at providing the student with a medium to perform live system analysis and gain tool-use experience. This research focused on introducing students to the eight categories/types of operating system artifacts he/she is most likely to encounter while performing a preliminary analysis of a WinOS machine that has exemplified odd/unusual behavior. The volatile artifacts are: Network Connections, Users Logged On, and Processes. The non-volatile artifacts are: Accounts, Files, Logs, Tasks, and the Registry.

The VMs ran full-fledged operating systems peppered with artifacts from actual exploitations conducted within a controlled lab environment. In the end, a malicious persona taking actions on a system will leave artifacts behind. The ability of first responders to seek, identify and analyze these artifacts is invaluable for detecting evidence of a CIRCE, for making timely and accurate reports, and ultimately, for developing the correct response actions to eradicate the threat.

B. FUTURE WORK

1. Classroom and Individual Use Evaluations

An in-classroom evaluation would provide good feedback to revamp and refine the labs provided. A study consisting of students with incident handling backgrounds from none to highly proficient would provide valuable feedback on how well the Lab and Virtual Environments can teach the desired skills. This should lead to improvements in the lab, which aim at creating a training aid detailed enough to teach a beginner, and challenging enough to provide refresher training to the more experienced incident handler.

2. Initial Response Actions

The work conducted in this research only touches lightly on the first two phases of the Incident Handling life cycle: Detection and Preliminary Analysis. It is very likely that a first responder will also be the individual tasked with implementing well-chosen

initial response actions that help to ensure the containment of any damage caused (realized), or likely to be caused (potential). This would transition the first responder out of Phase 2 and into Phase 3: Preliminary Response Actions. A compliment to the product of this thesis research would be follow-on lab-guided training concentrating on Phase 3 goals and tactics.

3. Development of Lab in a Game-Like Environment

Development of the Challenge Lab into a more interactive game would add tremendous value to the training experience. The game would allow for the player to be involved in organization set up such as what equipment to use and what policies to put in place on software such as antivirus and firewalls. This would allow for development of a more realistic network with the potential of injecting or removing artifact based on the user's actions.

4. Mobile Application and Web Browser Artifacts

The amount of available artifacts can differ from system to system and version to version. This research complied the type of system artifacts that are most commonly found across multiple platforms. However, these are by no means the only artifacts that are viable for first responder investigation. Given the evolving arsenal of user-friendly tools, a further study into common operating system artifacts and associated investigation tools that help develop the incident picture would be beneficial. For instance, in the world where Bring your Own Device (BYOD) is becoming a common business practice, a follow-on study of the effectiveness of these artifact in such an environment is would be beneficial.

5. Employ Later Windows OS Versions in the Virtual Environment

In time, the already aged Windows XP will be phased out of use by U.S. Naval systems. Creating VMs based on Windows 7, 8, 10, or later versions will eventually be needed to educate students about artifacts in use on those—current and future--systems. There is always a possibility a new class of artifact will arise with advances in OS and other programs.

THIS PAGE INTENTIONALLY LEFT BLANK

APPENDIX A. TRAINING LAB

Artifact Investigation Training Lab
Outline by LT Tye Wylkynsone, LT Simone Mims

Lab Objectives:
This lab will allow students to employ investigative techniques and methodology to analyze available artifacts. Students will learn to complete and submit a thorough incident report.

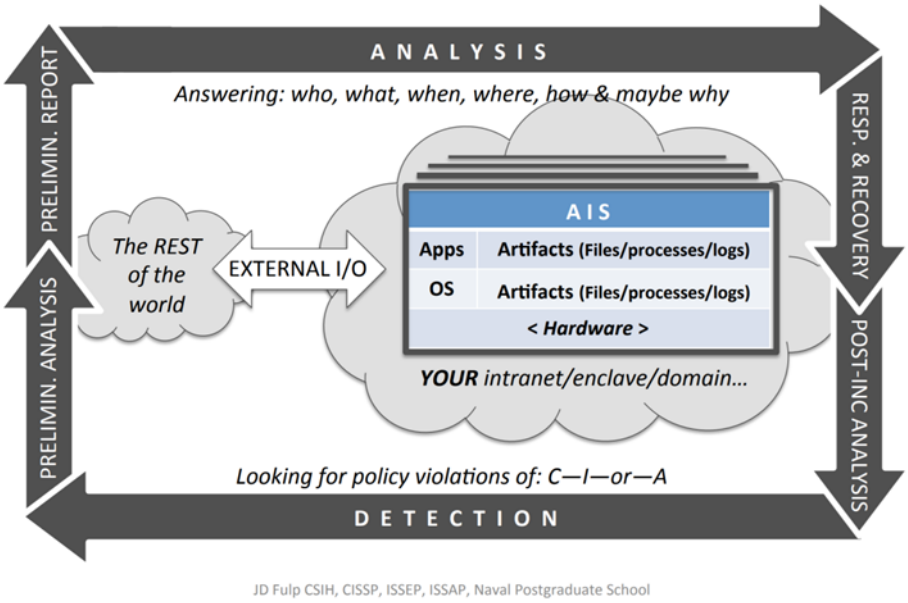


Figure A1. 1CJCSM Incident Management Process Life cycle Model

Discussion:

The Department of Defense Incident Handling process is broken into the six phases depicted in Figure 1. As an individual who may discover or be called upon to investigate a potential incident, a basic understanding of operating system artifacts and tools used to investigate them will help you be a more effective first responder.

Upon detection of a suspicious event, the first responder should immediately move into the preliminary investigation phase, which includes but is not limited to searching for artifacts and querying users. During this phase, the investigator will determine if the event should be considered a reportable event or incident based on a standardized benchmark (6). Timely and concise reports ensure the information gets to the right people needed to defend and restore the system.

Your role: Approach this lab from two perspectives. Start as an everyday user, then transition into your role as the Incident Responder (IR).

Instructor role: I am the organization's "Admin," and I am your **only** POC for amplifying information.

Amplifying information: For purposes of this lab, some volatile artifacts are simulated to give the student the opportunity to practice with the suggested tools. This will be clearly marked in the lab.

Tasks:

T1. Set up Virtual Environment

1. Obtain a copy of the Training Lab from the instructor as an .ovf folder.
2. Import this into the Virtual Machine software of your choice (e.g. VMWare Workstation or VirtualBox, etc.)
3. Launch the VM.

Both the everyday user and the Incident Responder should always have a detection-phase mindset. In order to recognize that something suspicious is taking place, every organization should have some written guidance and basic understanding of what normal looks like for their network.

Q1. What are some good practices that should be in place on any network in order to ensure everyone in your IT organization understands what "normal" is and to allow proper detection to take place?

A1. System backups, Written guidance to both users and sysadmins, Well-defined firewalls, antivirus systems, Intrusion detection/prevention systems, training on what users can do to prevent things like phishing and malicious downloads etc. Read-only tool kit that is not on the system a system. Written Incident Response Guidance, Defined First Responder tasking, saved system baseline information

Besides good written policies and a robust backup plan, another good practice that should be in place in any organization is the development of an investigation tool kit. Since an attacker could manipulate tool names and functionality, a tool kit will ensure that *known good* applications are being used for investigation and will have no unintended effects on the system.
According to CJCSM 6501.01b, event detection may occur in a number of ways. For instance, an automated detection system or sensor, a report from an individual or user, or an incident report or situational awareness update from other internal or external organizational components, such as USCYBERCOM, US-CERT, or external Computer Security Incident Response Team entities (CJCSM 6501.01B B-9). Review page B-9 and

B-10 of the CJCSM 6501.01b to see examples of suspicious cyber events and various ways they are detected.

T2. Recognize the Initial Signs and Warnings.

Figure A2. 2CJCSM Incident Management Process Life cycle Model

Q2. Are there any changes to the login screen that appear unusual to you? What should your next action be?

A2. A new account has been created, pwnedU, we should contact our network administration team

Now that suspicious behavior has been detected, move out of the detection phase (I) and into the preliminary investigation phase (II) of the CJCSM Incident Handling Life cycle model. Transition to the Incident Responder role here.
One way a first responder can categorize artifacts is by their volatility. A prudent place to begin one's investigation is to focus on those artifacts that have the shortest lifetime or duration: Volatile Artifacts. In this lab, the following artifacts are considered volatile: Network connections, Users currently logged on, Processes, and currently-scheduled Tasks.

T3. Print the Reporting form on page 13–15 of this lab. Fill-in as much information as possible for the **Cyber Incident Tracking Informatio**n and **Reporting Information** section of your report.

Appendix B to Enclosure C of the CJCSM 6510.01B provides a Cyber Incident Reporting format to utilize when making your initial report. This will be helpful in gathering and properly reporting the suspected incident or reportable event information. As mentioned above, your initial report is critical to getting the incident under control as quickly and efficiently as possible. Timeliness wins over completion here. Once the initial report has been made, analysis can continue and the report can be updated as more information becomes available. A well-formulated report will be as accurate and as succinct as possible. See page C-B-1 through C-B-5 of the CJCSM 6501.01b for a future guidance/explanation of what information is desired for each report block.

T4. Login to the Georgia account with the password **"password"** and begin investigation of available system volatile artifacts. Investigation tools have been suggested for quick analysis of each artifact however there are many other tools that could be used to gather information. If you have a tool you know better, feel free to use it.

Accounts is a non-volatile artifact, but we begin with it because it was the first-discovered type of artifact.

A) Accounts

An account is a collection of permissions and settings. An operating system uses it to grant resources to personas who authenticate themselves correctly for a particular account. Should an attacker have physical access to a system, they may be able bypass the account system to gain illicit access to their target. However, it is far more likely that the attacker will attempt remote access to a system. The OS guards against any access not involving an account.

T5. From the start menu open a command prompt, type the command **net user** to view all accounts on your system.

> **Q3.** What account listed requires further investigation? Describe what you find by a quick look at the account(s) properties and privileges.
>
> **A3.** PwnedU was created on 17AUG and is part of the Administrators group

Further investigation of the suspicious account that we noticed at log-in, pwnedU, should have revealed that not only was this account recently created on the 17th of August but it is also part of the administrative group.

B) Network Connections

At this point in the investigation, network connections are a good follow-on artifact. Abnormal or unexpected connections could signal unauthorized remote access that requires further investigation. In this case, you may suspect that a remote connection was used to create the suspicious admin account, pwnedU. This artifact is time-critical as

a connection must be established for you to see it. A skilled advisory will attempt to limit the time the connection is in use to prevent their own discovery.
Netstat, a built-in Windows command-line tool, is one tool that could be used to determine if any unexpected communication links are established with your system. If the attacker was active at the time of your investigation, this tool will potentially help discover where the remote connection originates.

T6. In the open command window, type the command ***netstat*** to view the active network connections**.** Since the connection has been terminated, you ought to see a blank output. Figure 3 captured the attacker's active connection. Use this screenshot to answer the questions in this section.

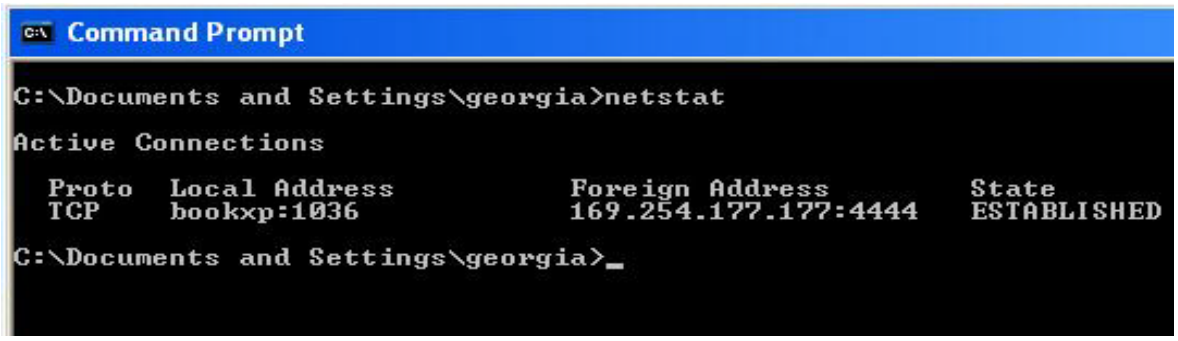


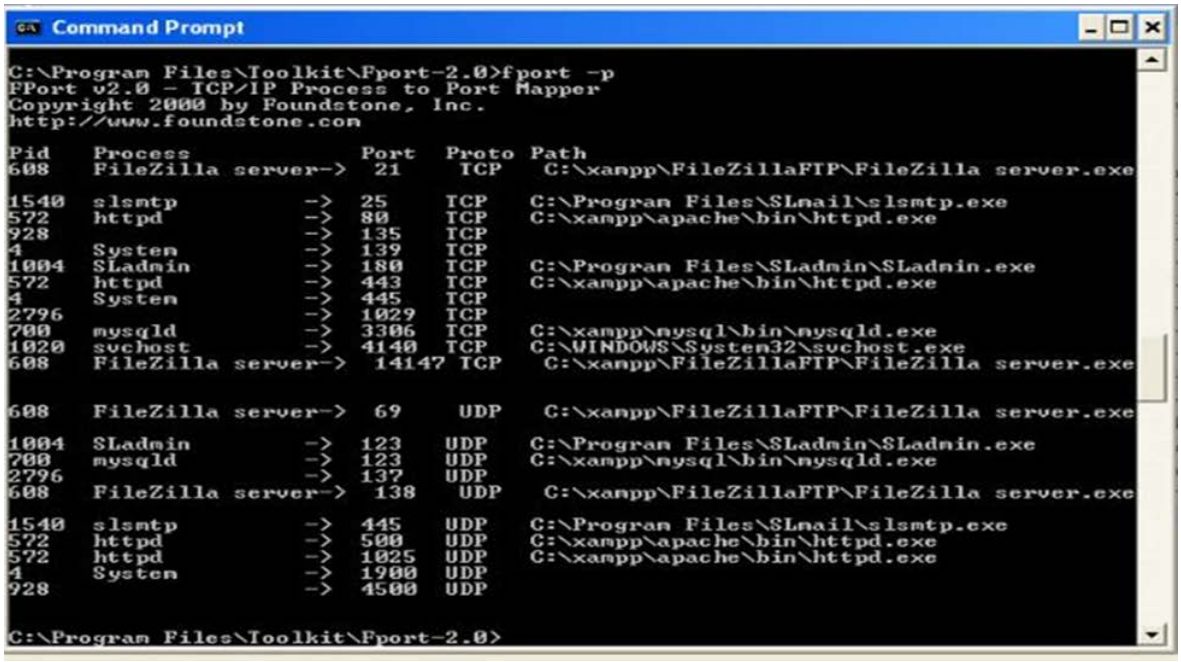
Figure A3. Active Network Connections

Notice a TCP session is open (or 'established') with an IP address you are unfamiliar with. You may or may not recognize the port number in use. Do a quick Internet search to learn more about what this port is commonly used for.

Q4. What is this port typically used for?

A4. W32.Blaster Worm and Trojan horses

T7. In the open command prompt run fport.exe. Navigate to **C:\Program Files\Toolkit\Fport-2.0** Once there, type the following command to run fport: ***Fport.exe*.** Your screen should look similar to the screenshot below.



```
Command Prompt
C:\Program Files\Toolkit\Fport-2.0>fport -p
FPort v2.0 - TCP/IP Process to Port Mapper
Copyright 2000 by Foundstone, Inc.
http://www.foundstone.com

Pid   Process              Port   Proto Path
608   FileZilla server->   21     TCP   C:\xampp\FileZillaFTP\FileZilla server.exe

1540  slsmtp          ->   25     TCP   C:\Program Files\SLmail\slsmtp.exe
572   httpd           ->   80     TCP   C:\xampp\apache\bin\httpd.exe
928                   ->   135    TCP
4     System          ->   139    TCP
1004  SLadmin         ->   180    TCP   C:\Program Files\SLadmin\SLadmin.exe
572   httpd           ->   443    TCP   C:\xampp\apache\bin\httpd.exe
4     System          ->   445    TCP
2796                  ->   1029   TCP
700   mysqld          ->   3306   TCP   C:\xampp\mysql\bin\mysqld.exe
1020  svchost         ->   4140   TCP   C:\WINDOWS\System32\svchost.exe
608   FileZilla server->   14147 TCP   C:\xampp\FileZillaFTP\FileZilla server.exe

608   FileZilla server->   69     UDP   C:\xampp\FileZillaFTP\FileZilla server.exe

1004  SLadmin         ->   123    UDP   C:\Program Files\SLadmin\SLadmin.exe
700   mysqld          ->   123    UDP   C:\xampp\mysql\bin\mysqld.exe
2796                  ->   137    UDP
608   FileZilla server->   138    UDP   C:\xampp\FileZillaFTP\FileZilla server.exe

1540  slsmtp          ->   445    UDP   C:\Program Files\SLmail\slsmtp.exe
572   httpd           ->   500    UDP   C:\xampp\apache\bin\httpd.exe
572   httpd           ->   1025   UDP   C:\xampp\apache\bin\httpd.exe
4     System          ->   1900   UDP
928                   ->   4500   UDP

C:\Program Files\Toolkit\Fport-2.0>
```

Figure A4. IP Processes to Port Map

As you may have noticed, the suspicious port is not in the list as the remote connection is no longer active. However, there could be other ports here that give indications of how the system was compromised.

Q5. What are some ports listed that you do not recognize? Do a little more research into these ports and their associated applications/paths. Write any observations below.

A5. 445 used by System and slsmtp. The slsmtp has been shown to be commonly used for buffer overflow attacks (sever vulnerability) 138 used by FileZilla. File Zilla allows direct access to website servers via FTP. Malware may often camouflage itself as FileZilla.exe. 123 used by mysqld which is the MySQL server Daemon this port is often used to complete NTP attacks or control implanted trojans.

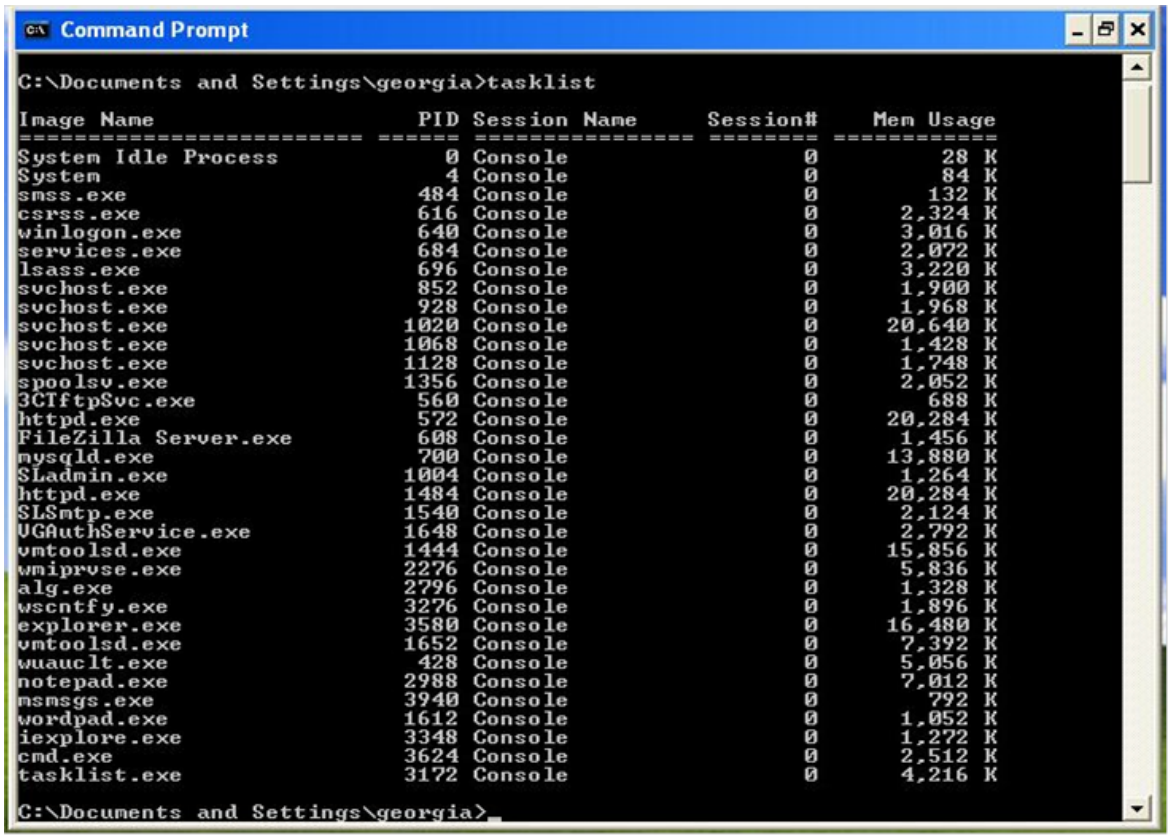
An Internet search of some of these well-known (<1024) ports will reveal associations with SMB (server message block -445) and NetBIOSover TCP/IP (NTP -139), a mySQL daemon (-123), and suspect FileZilla activity (-138). All these open port present serious security vulnerabilities which may be used by an attacker.

C) Processes

The next artifact that could be investigated are the processes. A process is required by any action that demands resources and causes the OS to begin running through a set of instructions. This includes actions taken by an attacker. The trail created by these artifacts may reveal rogue or abnormal behavior.

There are a few tools both resident to the WinOS and to your tool kit that will help in the analysis of processes. Tasklist and Autoruns, respectively, are utilized in this section. Feel free to branch out to other tools if you know them. Always keeping in mind having as light a touch as possible.

Immediately upon logging in, you ran the **tasklist** command to see what processes were running. The screenshot below is what you saw. The goal of looking at the list of running processes is to recognize if any running processes are out of the ordinary. This means you must understand what is normal for this system and the user. Useful information could also be gained from asking the user what they were doing during the suspected infection time and removing the known good processes from the list. In this case, you have asked the user who reported that they had an Internet browser window, WordPad and Windows messenger open.



```
Command Prompt

C:\Documents and Settings\georgia>tasklist

Image Name                     PID Session Name     Session#    Mem Usage
========================= ====== ================ ======== ============
System Idle Process              0 Console                 0         28 K
System                           4 Console                 0         84 K
smss.exe                       484 Console                 0        132 K
csrss.exe                      616 Console                 0      2,324 K
winlogon.exe                   640 Console                 0      3,016 K
services.exe                   684 Console                 0      2,072 K
lsass.exe                      696 Console                 0      3,220 K
svchost.exe                    852 Console                 0      1,900 K
svchost.exe                    928 Console                 0      1,968 K
svchost.exe                   1020 Console                 0     20,640 K
svchost.exe                   1068 Console                 0      1,428 K
svchost.exe                   1128 Console                 0      1,748 K
spoolsv.exe                   1356 Console                 0      2,052 K
3CTftpSvc.exe                  560 Console                 0        688 K
httpd.exe                      572 Console                 0     20,284 K
FileZilla Server.exe           608 Console                 0      1,456 K
mysqld.exe                     700 Console                 0     13,880 K
SLadmin.exe                   1004 Console                 0      1,264 K
httpd.exe                     1484 Console                 0     20,284 K
SLSmtp.exe                    1540 Console                 0      2,124 K
VGAuthService.exe             1648 Console                 0      2,792 K
vmtoolsd.exe                  1444 Console                 0     15,856 K
wmiprvse.exe                  2276 Console                 0      5,836 K
alg.exe                       2796 Console                 0      1,328 K
wscntfy.exe                   3276 Console                 0      1,896 K
explorer.exe                  3580 Console                 0     16,480 K
vmtoolsd.exe                  1652 Console                 0      7,392 K
wuauclt.exe                    428 Console                 0      5,056 K
notepad.exe                   2988 Console                 0      7,012 K
msmsgs.exe                    3940 Console                 0        792 K
wordpad.exe                   1612 Console                 0      1,052 K
iexplore.exe                  3348 Console                 0      1,272 K
cmd.exe                       3624 Console                 0      2,512 K
tasklist.exe                  3172 Console                 0      4,216 K

C:\Documents and Settings\georgia>
```

Figure A5. Running Process

T8. From the command prompt line, type in **tasklist/svc** to display all running processes. Compare what you see now with what you saw when you first ran the command as well as with the information provided by the user.

> **Q6**. Based on the Figure 5 list and what was discovered from the fport results, do you see any processes that could be suspicious?
>
> **A6**.-notepad, SLadmin.exe, SLsmtp.exe , mysqld.exe , fileZIlla

Any process related to (or matching the name of) one of the suspect programs utilizing a port mentioned in A5 is a noteworthy artifact. Further investigation of these processes can be conducted with ProcMon, a SysInternals tool for passively monitoring process activity on windows systems.

T9. From the command prompt, navigate to **C:\Program Files\Toolkit\SysInternals Suite**. Type **ProcMon** and hit <enter.> Below is a process tree (available under the Tools menu) containing volatile information.

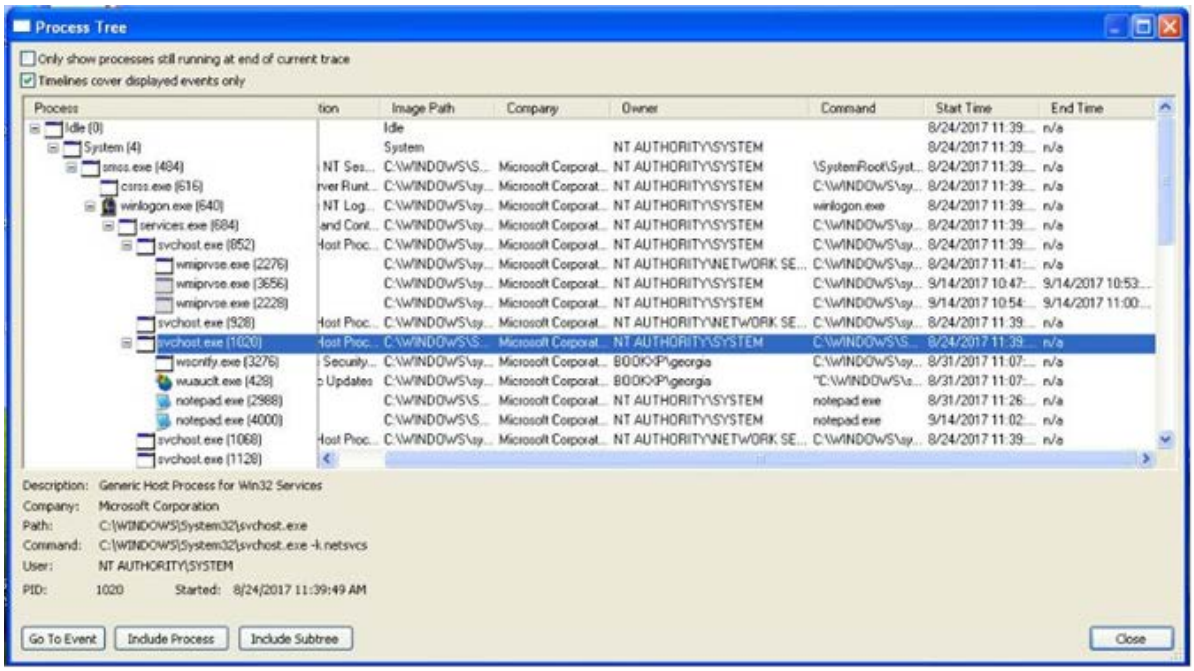


Figure A6. Process Tree View in PROCMON 3.1

The *First Responder's Guide to Computer Forensics: Advanced Topics* provides 19 checks to conduct that could be useful in helping you identify if the legitimacy of processes (Nolan et al., 2005).

T10. Run the suspicious process (Slsmtp) in A6 through the 19 checks listed below. Make this a brief mental exercise as data may unavailable for answering some of these questions. Note: PROCMON is useful to answer most of the questions. There are multiple correct answers to the questions below. Suggested answers are provided.

1. Process' Name _~~Slsmtp.exe~~___
2. Process' Extension __~~Exe~~_____
3. Location of Process' Image (File) _~~C:\Program Files\Slmail\Slsmtp.exe~~_____
4. Process' Parent __~~Service.exe~~_____
5. Process' Children ___~~none~~_____
6. Number of Process Instances ___~~1~~_____
7. Process' User/Owner Account __ ~~NT Authority/System~~ _____
8. Process' Start and Elapsed Time ___~~May differ per student based on when they look~~_____
9. Process' Command Line Arguments ___~~none~~_____
10. Process' Base Priority ___~~8/Normal~~_____
11. Network Connections/Ports Created by Process _____~~445~~___
12. Interesting Strings found in Process' Image ___~~none~~_____
13. Process' Current Execution Status ____~~Running~~____
14. Process' Hash (as compared to known good reference) _____
15. Process' Existence When Not Expected (or opposite) ___~~N/A~~_____
16. Process' Resource Utilization ____~~0/.0625~~____
17. Files Opened and Privileges Thereof (List One) *use handles.exe* __ r_____
18. DLLs Loaded by Process(List Two) *Double-click the suspicious process in PROCMON and click the process tab or utilize listdlls.* ___~~AntiSpam.dll and SMTPLog.dll~~___
19. Process' Persistence ___~~Exists~~_____

Q7. Determine if these processes were legitimate or rogue. Could it be malicious?

A7. Any process cited in A5 will turn up a port known for malicious activity. So, that answer carries over to here. Also, researching these ports and the answers in A6 all answer Q1: names that match known malware or exploitable programs. However, the Fig. 6 shows Notepad.exe with two running instances, both owned by the System (Q7). These were not scheduled nor were they used by Georgia, but by the attacker impersonating 'System.'

Process questions 7 and 15 highlight how some children of 1020 have been started by the System, not the user. Yet the user has not employed that program and it was not a

scheduled task either. Also, question 11 has already been used (during Q6) for all the programs that opened ports known for their malicious activity.

D) Users currently logged on

It is a good idea to check which users are currently logged on to the system. It is possible that our attacker is currently logged on.

T11. From the command line, navigate to **C:\Program Files\Tool kit/SysInternals Suite** Type the following command to view users currently logged on**:** ***psloggedon.exe*** and hit <enter.>

Q8. Which users are currently logged on?

A8. Georgia and today's timestamp

It is possible that not every artifact will provide you with a smoking gun signifying clear intruder behavior. Do not get stuck trying to make evidence appear. Remember this is the preliminary investigation phase and your goal is to provide as clear and ***timely*** a report as possible.

E) Scheduled Tasks

The attacker may have set up malicious activity to be triggered by some event. Scheduled Tasks represent one overt way of doing this. These tasks are orders to the OS to run a chosen program in conjunction with a selected event. Any artifact left here by an attacker may indicate one of their goals for exploiting the system.

T12. From the open command prompt, ensure you are still in your toolkit folder and type ***autoruns.exe.*** Once the application opens, click on the ***scheduled task*** tab to view any current or future scheduled task. This information can also be listed from the command line with the ***schtasks***. Try these methods and answer the question below.

Q9. What suspicious tasks are currently scheduled? Give as much detail about the scheduled task(s) as possible. [Right click any scheduled task that requires further investigation and select ***Jump to Entry***. This will pull up this scheduled task in the Windows Task Scheduler GUI. Once you are in the Windows Task Scheduler, right-click and select ***properties*** to learn more about that task.]

A9. LaunchTrojan.job set to launch an .exe file called BadRat, itself set to run at a specific date in the file and delete the task once it has run.

You now have discovered evidence that an incident has occurred. In accordance with your role as preliminary investigator, it is important to report the incident and what information you have gathered. Remember you are in the preliminary analysis of the investigation phase and should not search exhaustively for evidence after you have determined a reportable event or incident has occurred. For the purposes of the lab, continue investigating other artifacts.

T13. Update the CJCSM report**.** Use the information you gathered in Task 1–10 to fill in the following blocks of your report: Primary incident category, delivery vector, source IP(s) and ports, Target IP and ports, Method of Detection.

Preliminary investigation of the volatile artifacts is now complete and you should move on to the nonvolatile artifacts. Artifacts which fall into the nonvolatile category are: Accounts, Logs, files, and registries.

F) Logs

There are numerous logs that are recorded on a host machine daily. Each can be a great tool for rebuilding the story of what took place during a suspected incident. Based on how the logs in your system are configured, there could be hundreds of thousands of logged events to sift through. Utilizing what information was gathered about the incident so far, an attempt can be made to scope the incident and focus your investigation of the logs into a particular time period. Unusual or missing logged events can also be great indicators of where you should look next.

Q10. *What are some basic logs that are available on virtually all systems that should be examined during the preliminary investigation phase?*

A10. Security, System, Application

Q11. *What are 5 good activities to look for while investigating logs?*

A11. Failed log-on attempts, started or stopped services, remote connections, created or deleted accounts, started or stopped processes.

T14. Launch the Windows Event Viewer by typing **eventvwr.msc** into a command prompt window. Navigate to each of those core logs you entered in A10 above.
PSloglist is a SysInternal Suite command line tool that allows you to view log data in a human-readable output or to dump logs into a plaintext (Excel) file for a searchable review capability.

T15. From the command prompt window, navigate to **C:\Program Files\Tool Kit\SysInternals Suite** and type the desired PSloglist command to continue your log investigation. Use this website to learn more about using this tool: https://docs.microsoft.com/en-us/sysinternals/downloads/psloglist
Pay special attention to any of the helpful events you listed above or that you may have heard in class. These should include successful and failed logon attempts, creating/stopping/starting services, usage of suspicious applications, changes to account and security settings, changes to user permissions, and Event IDs.

Q12. *What useful information did you identify in the logs based on the information you have gathered in your investigation so far? Write below what you found and where you found it.*

A12. When pwnedU was created (17AUG17) there is much activity by SLadmin. Multiple errors can be found though none seem out of the normal. Also, if you investigate the 7035 Event ID from 2017 from the dates around pwnedU creation, you will find the starting of the remote connection manager this may be a great way to develop a timeline of the attacker's presence on the system. (Answers may vary here; the point here to get the student looking through the logs and filtering out any behavior that could help their investigation.)

Because so many events are logged on daily basis, it is important to filter the logs with the information you have already learned. This will direct you to the most useful portion of the logs. In this case, you know a few things already: You have identified some suspect, running programs; you know an account was created; you suspect this account was created through a remote connection (service was started) and you know when that account was created, August 17th. Use this date and program information to focus your investigation and look at logs that are a month before and a few months after this date and logs associated with the suspicious programs.

Once you have determined a likely timeframe for the incident, select the log you want to view and further filter the list by selecting **View** on the menu bar and **Filter Current log.** Here you will be able to filter out unnecessary information to help narrow down your search. Some available filters are: Time filter, Event Source, Event ID, Task Category, Keywords, User, and Computers. Make sure to clear any filters you may have set as you move between logs by returning to the **View** menu and selecting **All Records.** A description of how to set each desired filter can be found at: https://technet.microsoft.com/en-us/library/cc722058(v=ws.11).aspx.

It is also important to note that a blank log can also be a red flag and an indication that the attacker may have been trying to cover-up his or her tracks. For instance, you may have noticed that the security logs are blank. Account logon events, Account management, policy changes, object access, successful or failed network connections, and privilege use, are a few of the events logged in the Security logs that could be helpful in rebuilding the attacker's story.

G) Files

Files are the means to store data for a computer or information system. When an attacker creates a file on your system (or uploads it), there is a high likelihood it has a malicious purpose. A file also becomes an artifact when an attacker leaves any trace on it. Perhaps they stole information, read sensitive data, or, or embedded an alternate data stream into a file. A file integrity management system is a great way to check for changes to your system, including files. Linux-based systems allow you to use programs such as AIDE and TripWire, which make use of a known good previous image to compare changes. A very basic approach to investigating a Windows file is to view the file properties.

Q13. By looking over the various files located anywhere in Georgia's computer, how many are there?

A13: 30920

If you are stuck here, try a simple windows search for files on the C:// drive.

Q14. Looking over the C-M-A timestamps for these files, are there some that are contemporary to the attacker's activity on the system? Or when the user was logged on?

A14: Date of badRAT file precedes creation of the unknown account by a week, so the attacker exploited the system by 10AUG17 or earlier.

As Q13 shows, there are over 30,000 files on the system. Use the information that you know about the most likely dates the attacker may have been on the system. Start investigating any files created or changed around that date. If you have the reporting user on hand, you can ask them about some of the candidates you turned up. Recall Georgia previously had not stated anything about working with Notepad.

Q15. What suspicious text files were found?

A15: BadRat.txt located in the system32 folder not a normal location for a .txt file.

You may have noticed that the suspicious .txt file was hidden in an unusual location for text files. This is likely the attacker's attempt to hide his actions. Now you have an ominously-named file, possibly created by a program that was not being used by Georgia. Tampering with such a file risks detonating it --- with unknown consequences. Leave that for the data forensics lab.

The SysInternals package provides some noteworthy tools for file investigation. 'Strings' looks for suspicious characters in files. 'Streams' checks for alternate data streams and provides a means for deleting them. 'PsFiles' shows all files opened remotely.

T16. Try using some of these commands to become familiar with potential output results. Run these programs from the command line as an administrator:

"streams *"

"strings -u -q *Executable_name.exe* | sort" Where –u is Unicode and –q is quiet mode. Select a file of interest.

"Psfile"

Q16: What output was provided? What information could be gained from these commands/tools?

A16: 0 files will be found (Take the opportunity to evaluate the different types of output you would see using these command.)

H) Registries

Registries are sets of values (known as configurations) that an Operating System requires for proper functioning. These values pertain to the system and for each user. This artifact should be investigated with caution as perturbing these values could cause user or system malfunction or breakdown.

The Registry database is broken into five major hives, or files, located at: %SYSTEMROOT%\system32\config and two user-specific hives located at the user's profile directory

T17. From the command prompt window, navigate to **c:\Windows\system32\config** and type ***dir <enter>*** to list the contents of the folder.

Q17. What are these five hives called? You may also find this information in the regedit application by expanding the HKEY_Local_Machine folder. To open this program, click the start menu's run, enter ***regedit*** and press <enter.>

A17. SAM, Default/Hardware, Security, Software, System

Native to the windows systems are both a command-line and GUI interface tool. Both can be used to view, query, and modify registries. The repository of information on usage of the system and applications is extensive. They could readily expose an exploitation to a (highly) knowledgeable technician. Through examination of the registries' repositories, an investigator may be able to gain information such as the last actions taken by a specific user on the system, which programs are set to auto-run, running applications, and hardware configurations. Any unusual findings may indicate malicious activity took place.

One place to begin your investigation of registries is the Auto start keys. These deal with programs that run during the startup of the system. Autorun.exe is a great tool for quick investigation of the auto-run Registry Keys. Here we will look at three different keys, the Run, RunOnce and WinLogon Userinit keys.

T18. From the command prompt enter any of the following commands for a quick view of each auto-run registry: (make sure your command prompt looks like C:\>)

> **C:\> reg query HKLM\Software\Microsoft\Windows\CurrentVersion\<Runonce, Run, RunonceEX>**
>
> *Ex.* reg query HKLM\Software\Microsoft\Windows\CurrentVersion\Run

For more information, utilize SysInternals tool 'autorun.' Once the application is launched, click on the tab labeled "Everything" to view the registry data. Look for the following three AutoRun Entries:

HKLM\SOFTWARE\Microsoft\Current Version\Run

HKLM\SOFTWARE\Microsoft\Current Version\RunOnce

HKLM\SOFTWARE\Microsoft\Current Version\Winlogon\Userinit

Ctrl+F can also be used to quickly locate entries with the keys words seen at the end of each entry above.

Further investigation can be done into these entries by right-clicking the description listed under the AutoRun Entry and selecting Jump to Entry to open windows registry editor. Or, Jump to Image will open the folder location of the file linked to a task. Include any information discovered in your answer below.

Q18. What suspicious entries, if any, did you find in these three registry locations?

A18. Though nothing particularly stands out during this search, the point is to get the students digging in the Registry in likely locations to look for traces of malicious behavior.

Attacker behavior may not always be plainly noticeable or any different than what you would expect to see in the registry for normal system operations. If nothing suspicious readily jumps out at you, remember you are in the pre-analysis phase. Realize that further analysis will still be conducted during phase IV of the Incident Handling process model.

T19. Complete CJCSM report.

Utilizing what you have learned in class, as well as the information gather from your preliminary investigation, complete the incident report and return it and this lab report to your instructor.

Modified version of Table C-B-1 from CJCSM 6510.01B (page 1 of 3) Reference pdf p.77 of document for guidance in filling in the report	
Field	**Description**
Cyber Incident Tracking Information	
Reporting Incident Number	
Organizational Tracking	
Incident Investigator Assigned	**PUT YOUR NAME HERE:** _____
Reporting Information	
Name	
Organization	
Telephone	
Email	
Fax	N/A (not applicable or not available)
Alternative Contact	jdfulp@nps.edu
Categorization Information	
Primary Incident Category	
Secondary Incident Category	
Delivery Vector(s)	
System Weaknesses	

Modified version of Table C-B-1 from CJCSM 6510.01B (page 2 of 3) Reference pdf p.77 of document for guidance in filling in the report	
Field	**Description**
Incident Status	
Status	
Incident Start Date	
Incident End Date	
Last Update	*<assume this is date of the assigned lab in course syll*abus>
Date Reported	
System Classification	
Action Taken	
Technical Details	
Event/Incident Description	
Root Cause(s)	

Modified version of Table C-B-1 from CJCSM 6510.01B (page 3 of 3) Reference pdf p.77 of document for guidance in filling in the report	
Field	**Description**
Source IP(s) & Port(s)	
Intruder(s) (if known)	
Origin (Country)	Fill in
Target IP(s) & Port(s)	
Technique, Tool, or Exploit Used	
Operating System (OS) and OS Version	
Use of Target (e.g., Web Server, File Server, Host)	
Method of [original] Detection	
Sites Involved	
Company/Organization	
Physical Location	
Network(s)	
Detecting Unit or Organization	
Affected Unit or Organization	
Impact Assessment	

Systems Affected	
Operational Impact	
Technical Impact	
The last eight entries in the original format were removed from this version as they are not germane to this exercise	

THIS PAGE INTENTIONALLY LEFT BLANK

APPENDIX B. TRAINING LAB REPORT

Modified version of Table C-B-1 from CJCSM 6510.01B (page 1 of 3) Reference pdf p.77 of document for guidance in filling in the report	
Field	**Description**
Cyber Incident Tracking Information	
Reporting Incident Number	786238
Organizational Tracking	NPS GZ35012
Incident Investigator Assigned	**PUT YOUR NAME HERE**:
Reporting Information	
Name	REPORTING USER
Organization	FNMOC
Telephone	555-3874
Email	user@nps.edu
Fax	N/A (not applicable or not available)
Alternative Contact	jdfulp@nps.edu
Categorization Information	
Primary Incident Category	Root User Intrusion
Secondary Incident Category	
Delivery Vector(s)	Trojan or Buffer overflow exploit on port 445 to SLsmtp.exe
System Weaknesses	Seattle Labs vulnerabilities are very old and well-known, leaving us open to an attacker. User account with too much privilege There may be a number of implants or other vulnerabilities placed on the system as a result of this intrusion.

Modified version of Table C-B-1 from CJCSM 6510.01B (page 2 of 3) Reference pdf p.77 of document for guidance in filling in the report	
Field	**Description**
Incident Status	
Status	Incident determined … Response and Forensics needed.
Incident Start Date	17AUG17
Incident End Date	25SEP17 --- and ongoing.

Last Update	*<assume this is date of the assigned lab in course syll*abus>
Date Reported	25SEP17
System Classification	UNCLASS
Action Taken	Preliminary investigation uncovered several indications of Root Intrusion and system exploits stemming from it. Quarantine of affected machines to allow for further deep investigation.
Technical Details	
Event/Incident Description	Buffer overflow exploit on port 445 to SLsmtp.exe
Root Cause(s)	Seattle Labs vulnerabilities are very old and well-known, leaving us open to an attacker. Georgia's account was hacked and the attacker gained Admin access. Extent of exploit needs urgent investigation.

Modified version of Table C-B-1 from CJCSM 6510.01B (page 3 of 3)

Reference pdf p.77 of document for guidance in filling in the report

Field	**Description**
Source IP(s) & Port(s)	162.255.177.177 ports 4444 and 445
Intruder(s) (if known)	pwnedU and whomever controls above source IP.
Origin (Country)	USA
Target IP(s) & Port(s)	Local. Bookxp:1036
Technique, Tool, or Exploit Used	Remote exploit of a buffer overflow to gain full access to our system. Possible Malware was installed and other vulnerabilities may have been created.
Operating System (OS) and OS Version	Windows XP Pro
Use of Target (e.g., Web Server, File Server, Host)	Terminal for SysAdmins, specifically Georgia.
Method of [original] Detection	User discovered unusual account.

Sites Involved	
Company/Organization	FNMOC
Physical Location	Naval Research labs, Monterey, CA
Network(s)	bookxp
Detecting Unit or Organization	Local IT at FNMOC and NPS ITACS.
Affected Unit or Organization	Entire FNMOC, possibly others.
Impact Assessment	
Systems Affected	UCLASS network at FNMOC
Operational Impact	Unknown.
Technical Impact	System may have numerous vulnerabilities added from this intrusion.
The last eight entries in the original format were removed from this version as they are not germane to this exercise	

THIS PAGE INTENTIONALLY LEFT BLANK

APPENDIX C. CHALLENGE LAB

Artifact Investigation Challenge Lab
Outline by Tye Wylkynsone, LT Simone Mims

Lab Objectives: This lab will allow students to employ investigative techniques and methodology to analyze available artifacts. This lab is meant to build on skills learned through completion of the Training Lab. Students will learn to complete and submit a thorough incident report.

Discussion: Detection of suspicious events can occur in multiple ways. Three methods outlined in the CJCSM 6501.01B are:

1. An automated detection system or sensor.
2. A report from an individual or user.
3. An incident report or situational awareness update from other internal or external organizational components, such as USCYBERCOM or US-CERT.

Recall from the Training lab that a responsible system user is always in the mindset of a detector. The detector is defined in the CJCSM 6501.01B as the individual who observe an event or incident. The detector should be trained to step away from the effected system and call for the organizations designated first responder. This will help to prevent damage or contamination of evidence. Once the first responder is on-scene triaging of the suspected event should occur as quickly and efficiently as possible. Good documentation and note taking will be key to successful completion of this lab. Some suggested notetaking templates are provided in the aids section of this lab. This lab is designed to build on what you learned the Training Lab. The “Company Network” has been built up to add more realism to the training.

Your role: You are entering work on a Monday after a well-deserved vacation. Remember to always be in the mode of the detector. Once a suspected CIRCE (Cyber Incident or Reportable Cyber Event) is detected, transition to the role of The Company’s first responder. Analysis Checklist and Aid have been included at the end of this lab to help organize your investigation efforts.

Instructor role: I am the organization's "Immediate Superior", and I am your only POC for amplifying information.

THE COMPANY

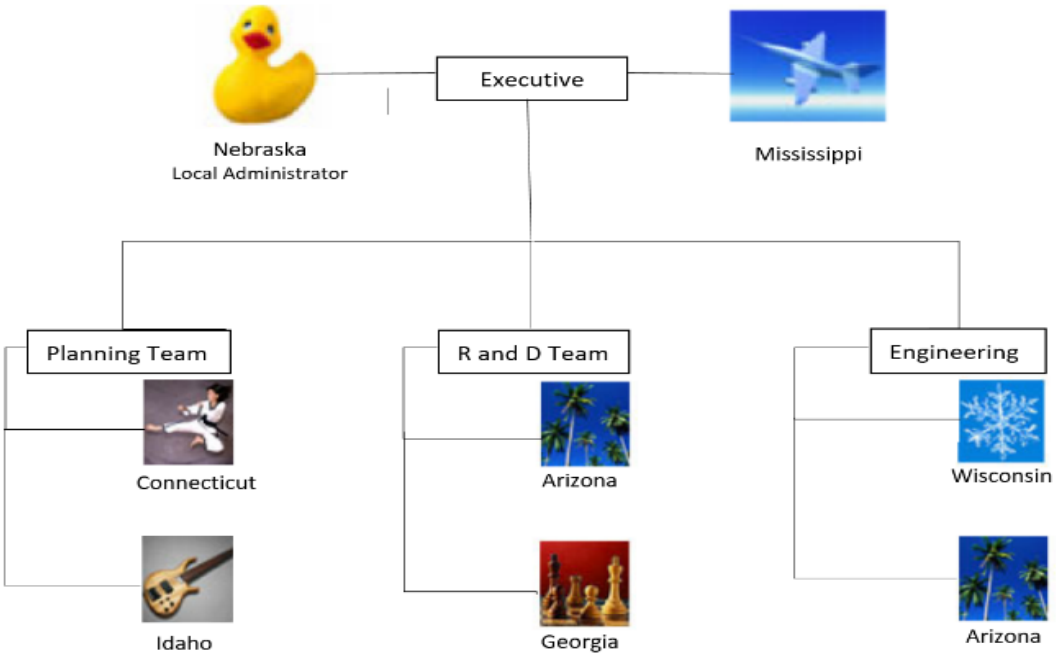


Figure A7. Company Organization Chart

Tasks:

T1. Set up Virtual Environment

1. Obtain a copy of the Challenge Lab VM from the instructor as an .ovf folder.
2. Import this into the Virtual Machine software of your choice (e.g. VMWare Workstation or VirtualBox, etc.)
3. Launch the VM.

T2. Log in as any member listed in the organization chart as a member of the Engineering group. Passwords for any account is the username spelled backwards. *Example: username: Wisconsin password: nisnocsiw*

Q1. Which account did you sign in as? What are the account properties? (Groups, standard or admin user, password expiration date, creator, etc)

A1.

Normal business can and should still occur simultaneously with the detector responsibilities. This also includes ensuring good security practices. Such as verifying your password is up-to-date and you are in accordance with company policies. This also includes acknowledging (i.e. press enter) all security or antivirus alerts and immediately reporting them to your SysAdmin or designated first responder.

T3. Check your outlook email box to see any new tasking orders. Navigate to the company share drive (Shared Documents) and review the System Use guidelines (Business Guidelines folder). While there, take note of what folders you can and cannot access based on your user accounts position in the organization.

In the process of going about your normal daily routine. You notice your computer is running a little slower than normal.

T4. Open a command prompt window and run the **tasklist** command. Note the processes that are currently running and list a few in the space provided below.

Running Processes:

_____ _____ _____ _____

T5. From the command prompt run the **net user** command and take note of how many accounts are available on the system. List the accounts below.

_____ _____ _____ _____

_____ _____ _____ _____

_____ _____ _____ _____

T6. Log out of the account you are currently logged in as and Log into one of the Executive accounts.

T7. Remember to note anything you feel is suspicious and if you suspect a CIRCE transition to the role of First Responder. An Artifact Investigation Aid is provided to assist in your analysis. Utilize the Cyber reporting form from the Training Lab to report your finding. When you feel you have filled out the report with enough information to inform your chain of command of what has occurred, complete the report and forward on to your Immediate Superior (i.e., the instructor). Note: Utilize no less than two artifacts to support your report claims.

Q2. What suspected CIRCE triggers you to transition to the first responder role? Give as much detail as possible about the suspected CIRCE. (This information can also be included in checklist 1)
A2.

Helpful checklist

Checklist 1

CIRCE Determination Checklist (answer what questions you can to help determine if the suspected event should be considered a CIRCE)

1. General description of the problem, event, or activity.
   _____
   _____
   _____
   _____
   _____
   _____
   _____
   _____
   _____
   _____
   _____
   _____
   _____
   _____
   _____
   _____

2. Status (ongoing or ended; successful or unsuccessful).
   _____

3. Number of ISs affected. _____

4. Source and destination Internet Protocol (IP) addresses.
   _____

5. Source and destination ports.
   _____

6. Hostname(s).
   _____

7. IS location.
   _____

8. User Information.
   _____

9. Timestamps.
   _____

10. IDS alert and payload (if relevant).

_____

11. Other.

_____
_____
_____
_____
_____
_____
_____
_____
_____
_____
_____
_____

Checklist 2
Process Legitimacy Determination Checklist

1. Process' Name _____
2. Process' Extension _____
3. Location of Process' Image (File) _____
4. Process' Parent _____
5. Process' Children _____
6. Number of Process Instances _____
7. Process' User/Owner Account _____
8. Process' Start and Elapsed Time _____
9. Process' Command Line Arguments _____
10. Process' Base Priority _____
11. Network Connections/Ports Created by Process _____
12. Interesting Strings found in Process' Image _____
13. Process' Current Execution Status _____
14. Process' Hash (as compared to known good reference) _____
15. Process' Existence When Not Expected (or opposite) _____
16. Process' Resource Utilization _____
17. Files Opened and Privileges Thereof _____
18. DLLs Loaded by Process _____
19. Process' Persistence _____

Artifact Investigation Aid		
Artifact	**(Sug.) Investigation Tool**	**First Responder Notes**
Process	***Note: see checklist 2*** Windows Command (arg): tasklist (/svc) SysInternals Suite tool: PROCMON	
Users Logged on	Windows Command (arg): SysInternals Suite tool: psloggedon.exe	
Files	Windows Command (arg): WinOS Search function SysInternals Suite tool:	
Network Connections	Windows Command (arg): netstat SysInternals Suite tool:	
Task (Scheduled)	Windows Command (arg): schtasks SysInternals Suite tool: autoruns.exe	
Accounts	Windows Command (arg): net user SysInternals Suite tool:	
Logs	Windows Command (arg): eventvwr.msc SysInternals Suite tool: PSloglist	
Registries	Windows Command (arg): reg, regedit.exe SysInternals Suite tool: autoruns.exe	

THIS PAGE INTENTIONALLY LEFT BLANK

APPENDIX D. CHALLENGE LAB REPORT

Modified version of Table C-B-1 from CJCSM 6510.01B (page 1 of 3) Reference pdf p.77 of document for guidance in filling in the report	
Field	**Description**
Cyber Incident Tracking Information	
Reporting Incident Number	786239
Organizational Tracking	NPS GZ35013
Incident Investigator Assigned	**PUT YOUR NAME HERE**:
Reporting Information	
Name	REPORTING USER
Organization	The Company
Telephone	555-3874
Email	user@nps.edu
Fax	N/A (not applicable or not available)
Alternative Contact	jdfulp@nps.edu
Categorization Information	
Primary Incident Category	Root User Intrusion
Secondary Incident Category	Malicious Logic or Non Compliance Activity
Delivery Vector(s)	Trojan or Buffer overflow exploit on port 445 to SLsmtp.exe
System Weaknesses	Seattle Labs vulnerabilities are very old and well-known, leaving us open to an attacker. Poorly named files in file system

	User account with too much privilege There may be a number of implants or other vulnerabilities placed on the system as a result of this intrusion.

Modified version of Table C-B-1 from CJCSM 6510.01B (page 2 of 3) Reference pdf p.77 of document for guidance in filling in the report	
Field	**Description**
Incident Status	
Status	Ongoing
Incident Start Date	6DEC17
Incident End Date	
Last Update	*<assume this is date of the assigned lab in course syll*abus>
Date Reported	Today's Date
System Classification	Unclassified
Action Taken	Preliminary Investigation of Logs and File system utilizing Windows Event Manager and File Search function respectively revealed newly created privileged account and newly created files and remote connections not initiated by the user. Recommendations have been made to take the disable reporting account and temporarily remove system from network pending further investigation.
Technical Details	
Event/Incident Description	Multiple user reports of request to change password messages. Users in the Executive Group reported multiple security alerts which suggest that were received after unusual behavior from the calc.exe application.
Root Cause(s)	Poor passwords and incorrect account types which allowed the attacker to utilize the buffer overflow vulnerability to get into the network.

Modified version of Table C-B-1 from CJCSM 6510.01B (page 3 of 3) Reference pdf p.77 of document for guidance in filling in the report	
Field	**Description**
Source IP(s) & Port(s)	196.254.33.45
Intruder(s) (if known)	
Origin (Country)	Source is Link Local address
Target IP(s) & Port(s)	169.255.197.17
Technique, Tool, or Exploit Used	Remote exploit of a buffer overflow to gain full access to our system. Possible Malware was installed and other vulnerabilities may have been created.
Operating System (OS) and OS Version	Windows XP Pro
Use of Target (e.g., Web Server, File Server, Host)	Terminal for SysAdmins, specifically Georgia.
Method of [original] Detection	Security and Antivirus Alerts.
Sites Involved	
Company/Organization	The Company
Physical Location	Monterey, CA
Network(s)	bookxp
Detecting Unit or Organization	Local IT
Affected Unit or Organization	The Company, possibly others.
Impact Assessment	
Systems Affected	UCLASS network
Operational Impact	Loss of Confidentially of Company Secrets and R&D results.

Technical Impact	System may have numerous vulnerabilities added from this intrusion. Password Security and Experimental data Results are Compromised
The last eight entries in the original format were removed from this version as they are not germane to this exercise	

A. ATTACKER'S STORYLINE

The attacker's ultimate goal was to collect as many passwords as possible through an installed key logger and steal/exfiltrate company experimentation results located on the company share drive. The Attacker installed a dummy script to distract the system users from the scripts that are accomplishing the goals outlined above.

B. ACTIONS BY ARTIFACT

Processes: Script is implanted, which opens and closes the calculator program 300 times. This triggers an alert message to be displayed to the student. The alert message reads as follows: "Malicious activity has been detected." If the student utilizes PROCMON before the script stops, they will see a process tree that has the calculator program as a child of wscript who is a child of the explorer.exe. This should be considered suspect as only the system administrator is allowed to run scripts on the system. Because this script is in the start-up folder, it will run every time the student logs onto the system. This is meant to demonstrate malware persistence.

Files: Script named FindCopy is implanted in the startup folder of both executive members. This script runs in conjunction with the calculator open-close script described above. The two run simultaneously in hopes that the very visible calculator behavior will distract users from the actions of the find and copy script. FindCopy finds the folder containing the company experimental data results and simulates reading and coping the data contained in these files to a new file which is created in the created by the adversary. A keylogger has also been implanted to steal account passwords. Each time the keylogger runs, a file is created in the attacker's folder to hold that password. An investigation into times that files were created and by whom will reveal that at least two folders were recently created by the signed in user (suspicious because the user did not report creating folders). There will also be evidence that information has been added to these folders.

Logs: Logs are replete with data for the student to filter and analyze. In the security logs, the student can find evidence of the Massachussets [sic] creation, as well as its privilege escalation and addition to the executive group. Time scoping will also reveal that the Georgia admin account has multiple failed login attempts and then a successful login right before the Massachussets [sic] account was created. A log entry is also made to the system log each time the find and copy script runs. The description of the event reveals

what ip address the supposed remote connect is established with and signifies to the student that a file transfer has occurred.

Registries/Accounts: In the HKLM Run key, a script has been added to the key value. This script runs when any user logs in and its goal is to delete the attacker's account (Massachussets [sic]). When students log out of the standard user and into the executive account or run the command net user, they should notice that the Massachussets [sic] account has disappeared. This happens without anyone taking action to delete it.

Users logged on: Use of the SysInternal tool **psloggedon.exe** will reveal that there are other users logged on to the system. However, this information does not advance the analysis.

Network Connection: N/A

Task Scheduled: Students may feel the need to review scheduled tasks, as there is a process running that was not user-executed. However, there are no tasks scheduled with this GUI. Students must investigate the registry with **regedit** tool or the windows start-up file to find the implanted malware.

THIS PAGE INTENTIONALLY LEFT BLANK

LIST OF REFERENCES

[1] United States Computer Emergency Readiness Team, "Incident definition," Department of Homeland Security. [Online]. Available: https://www.us-cert.gov/government-users/compliance-and-reporting/incident-definition. Accessed October 16, 2017.

[2] Office of Management and Budget, "Annual report to Congress: Federal Information Security Modernization Act of 2014," Washington, DC, USA, 2016. [Online]. Available: https://obamawhitehouse.archives.gov/sites/default/files/omb/assets/egov_docs/final_fy_2015_fisma_report_to_congress_03_18_2016.pdf

[3] Office of Management and Budget, "Annual report to Congress: Federal Information Security Modernization Act of 2014," Washington, DC USA, 2017. [Online]. Available: https://www.hhs.gov/sites/default/files/fy_2016_fisma_report%20to_congress_official_release_march_10_2017.pdf

[4] R. Nolan, C. O'Sullivan, J. Branson, C. Waits, "*First Responders Guide to Computer Forensics*," Carnegie Mellon, Pittsburgh, PA, USA, 2005. [Online]. Available: https://resources.sei.cmu.edu/asset_files/Handbook/2005_002_001_14429.pdf

[5] P. Cichonski, T. Millar, T Grance, K. Scarfone, "Computer Security Incident Handling Guide," National Institute of Standards and Technology. August 2012. [Online]. Available: http://nvlpubs.nist.gov/nistpubs/SpecialPublications/NIST.SP.800-61r2.pdf

[6] *Cyber Incident Handling Program*, CJCSM Directive 6510.01B, Chairman of the Joint Chiefs of Staff, Washington, DC, USA, 2012, pp A-1, B-8-B-16,B-29, B-A4, App B to Encl C. [Online]. Available: http://www.jcs.mil/Portals/36/Documents/Library/Manuals/m651001.pdf?ver=2016-02-05-175710-897

[7] A. Shaffer, "Cyber effects delivery: Exploitation," Lecture presented at CY4710 Adversarial Cyberspace Operations, Naval Postgraduate School, Monterey, CA.

[8] "Desktop Operating System Marketshare Worldwide," Statcounter, GlobalStats. [Online]. Available: http://gs.statcounter.com/os-market-share/desktop/worldwide. Accessed Oct 1, 2017.

[9] O. Skulkin and S. D. Courcier, *Windows Forensics Cookbook*. Packet Publishing, 2017. [Online]. Available: http://techbus.safaribooksonline.com/book/operating-systems-and-server-administration/microsoft-windows/9781784390495

[10] R. Messier, *Operating System Forensics*, Waltham, MA, USA: Syngress, 2016. [Online]. Available: http://techbus.safaribooksonline.com/book/operating-systems-and-server-administration/microsoft-windows/9781784390495

[11] Fulp, J. D. (2017). D&I WinOS registry. Lecture presented at CS4684 Cyber Incident Response and Recovery course, Naval Postgraduate School, Monterey, CA.

[12] A Shaaban and K., Sapronov, Practical Windows Forensics, Birmingham, UK. Packet Publishing, 2016. [Online]. Available: http://techbus.safaribooksonline.com/book/networking/forensic-analysis/9781783554096/practical-windows-forensics/pr01_html

[13] SANS Institute, Intrusion Discovery Cheat Sheet v2.0. 2003, pg. 2–3. [Online]. Available: https://www.sans.org/media/score/checklists/ID-Windows.pdf

[14] J. Luttgens, M. Pepe, and K. Mandia: *Incident Response and Computer Forensics*. USA: McGraw-Hill Education, 2014.

[15] H. Saxena, "What is an SMB Port? What is Port 445 and Port 139 used for?" The Windows Club. 10 Jan 2017 [Online]. Available: http://www.thewindows club.com/smb-port-what-is-port-445-port-139-used-for

[16] S. Cheshire, RFC 3927. May 2005. [Online].Available: https://tools.ietf.org/rfc/rfc3927.txt

[17] "wscript.exe-Windows-based script host by Microsoft," Malware Removal Instructions. [Online]. Available:http://deletemalware.blogspot.com/ 2013/06/what-is-wscriptexe-and-how-to-remove-it.html Accessed Oct 26, 2017.

[18] TechNet, "Configuring audit policies," Microsoft. [Online]. Available https://technet.microsoft.com/en-us/library/dd277403.aspx Accessed Sept 15, 2017.

[19] V. Dhanunjaya, Collecting volatile and non-volatile data. March 26, 2016. Linkedin. [Online]. Available: https://www.linkedin.com/pulse/collecting-volatile-non-volatile-data-vuppala-dhanunjaya

[20] S. Baird, "What is a process?" Processmodel. [Online]. Available: https://www.processmodel.com/blog/what-is-a-process/. Accessed Oct 20 2017.

[21] P. Denning, "The profession of IT fifty years of operating systems." *Communication of the ACM*. Vol. 59 No.3, 30–32, March 2016. [Online]. Available: http://denninginstitute.com/pjd/PUBS/CACMcols/cacmMar16.pdf

[22] M. Russinovich and A. Margosis, “Windows core concepts” in *Troubleshooting with the Windows Sysinternals Tools* 6th ed., Washington: Microsoft Press, 2016. [Online]. Available: http://techbus.safaribooksonline.com/book/operating-systems-and-server-administration/microsoft-windows/9780133986549

[23] W. Gleen, “What is the service host process (svchost.exe) and why are so many running?” June 9, 2017. [Online]. Available: https://www.howtogeek.com/howto/windows-vista/what-is-svchostexe-and-why-is-it-running/

[24] S. Oyamo and F. Flores. “Investigating OS artifacts for potential indictors of compromise (IOC)” Capstone Project, Department of Information Science, NPS, Monterey, USA, 2017.

[25] D. Litchfield, “NGSSoftware Insight Security Research Advisory: Multiple Buffer Overflows in SLMail,” May 7, 2003. [Online]. Available: http://attrition.org/security/advisory/nisr/nisr-07052003a.slmail

[26] Microsoft, “PsLoggedOn” last update: June 2016. [Online]. Available: https://docs.microsoft.com/en-us/sysinternals/downloads/psloggedon

[27] Fulp, J. D, “D&I WinOS Basic Artifacts.” Lecture at Naval Postgraduate School, Monterey, CA, 2017.

THIS PAGE INTENTIONALLY LEFT BLANK

INITIAL DISTRIBUTION LIST

1. Defense Technical Information Center
   Ft. Belvoir, Virginia

2. Dudley Knox Library
   Naval Postgraduate School
   Monterey, California