Dynamic Analytics-Driven Assessment of Vulnerabilities and Exploitation

Hasan Cam[1], Magnus Ljungberg[2], Akhilomen Oniha[1], Alexia Schulz[2]

[1]U.S. Army Research Laboratory, [2]MIT Lincoln Laboratory

1 Introduction

Most sectors providing the underpinnings of modern society have come to critically rely on computers and computer networks to function properly. These sectors include public health, finance and banking, business and retail, media and telecommunications, national defense, along with more fundamental critical infrastructure such as electrical power, water utilities, and food distribution. Our deep reliance on these sectors makes them particularly attractive targets for attack [29]. Adversaries can leverage computer or network vulnerabilities to interfere with the proper functioning of American society.

Trusted networks are used by the Department of Defense (DoD), Law Enforcement and the Intelligence Community (LE/IC), as well as countless Business and Industrial Enterprises (BIE) to provide access to critical information and services that are vital to accomplishing their respective missions. Collectively, we will refer to the DoD, LE/IC, and BIE as the trusted network user base (TNUB). Vulnerabilities on these trusted networks confer opportunity to adversaries of the TNUB to interfere with mission execution. Adversaries can leverage vulnerabilities to gain unauthorized access to the trusted network or to prevent authorized users from having access, either of which can negatively impact the missions of the TNUB. The motivation of adversaries to exploit vulnerabilities on a trusted network can be classified broadly according to five major categories: (*i*) Espionage and Intelligence Gathering, (*ii*) Denial of Service, (*iii*) Data Corruption and Misinformation, (*iv*) Kinetic and Cyber-Physical Effects, and (*v*) Hijack of Asset Control. A specific vulnerability may enable one or more of these classes of

adversary operations. The extent to which an adversary can leverage the vulnerability to interfere with mission success depends on (*i*) which of these five categories the vulnerability may enable and (*ii*) the extent to which mission execution can withstand adversary activity in each category. Therefore, the inherent risk presented by a vulnerability is specific to each mission that is impacted.

A critical national need in support of TNUB missions is to augment the current capabilities of vulnerability assessment tools to realistically assess attacker access to existing vulnerabilities and to improve the ability of mission leaders and planners to triage which system vulnerabilities present the highest risk to mission assurance. This requires a dynamic approach to vulnerability assessment rather than a static approach, because the attacker posture and vulnerability access as well as the way the trusted network is being leveraged to accomplish the mission are both subject to significant variability in time. The inherent challenge to filling this national need is that the available data are constrained to a limited number of observable vantage points: vulnerabilities are collected at the host locations, and observations of network traffic are limited to a small number of centralized tap locations, whereas the non-local problem of the attacker access depends more globally on the network topology. It will be necessary to estimate the vulnerability and resulting mission risk on the basis of incomplete information, gathered from myriad sensor types deployed at strategic locations. There are several enabling technologies that will be critical for satisfying this national need, none of which is currently deployed or exists. First is technology to dynamically infer network topology and the interconnection of hosts. Second is the ability to use this information, in conjunction with existing scans and other observations such as network traffic capture, to assess the severity of a vulnerability in terms of its specific impact to a

particular mission or set of missions. To this end, technology will also be needed to assess which assets each mission is leveraging as a function of time.

As the vulnerabilities and attack surface of assets grow in complexity and size, threats and malware also grow more pervasive, while cyber sensors generate more data to be analyzed. Intrusions are often obfuscated to the extent that its traces and fingerprints are hidden within different types of data (e.g., intrusion detection system [IDS] alerts, firewall logs, reconnaissance scans, network traffic patterns, and other computer monitoring data) that are involved with a wide range of assets and time points. However, even a small organization's security operation center may end up dealing with an increasingly huge volume of daily data. Given the time constraints, service level agreements, and computational and storage resource constraints in the analysis of such data, we aim at first identifying and extracting high-quality data products describing cyber events from the raw data. The analysis and assessment of these high-quality data products can be performed more quickly and dynamically by requiring a smaller amount of time and computational resources. We address the questions of (*i*) how the raw data size of cyber events can be reduced significantly at close to real time and (*ii*) what effective methods can be used to detect and analyze the noisy data of intrusion and vulnerability detections and exploitations. To this end, we undertake a holistic approach of considering the size and analysis of intrusion data, together with the analysis of vulnerability data and exploitations, by investigating how the cyber events and processes of intrusions and vulnerabilities are detected, cross-correlated, analyzed, and assessed.

In answering the two questions above, we present why data analytics, machine learning, and

temporal causality analysis are considered essential components, and we show how they interactively function in very important roles. High-quality data products can be extracted from raw cyber data by pinpointing the specific assets and time instances involved with intrusions. We suggest to use temporal causality analysis of main cyber sensor observations and events including intrusion alerts, vulnerabilities, attacker activities, firewall and HBSS log data, and network traffic. Our premise is that if we know what vulnerabilities exist in the system and how these vulnerabilities can be exploited by intrusion, then we can develop a causality analysis diagram for cyber events, vulnerabilities, intrusions, and observations of attacker activities. This causality analysis narrows down the cyber data to be searched and analyzed, leading to a significant reduction in size and scope from the raw cyber data. This results in faster data analysis, less computational resources, and potentially more accurate results. This chapter presents how data analytics could potentially leverage vulnerability assessment and causality analysis of vulnerability exploitation in the detection of intrusion and vulnerabilities so that cyber analysts can investigate alerts and vulnerabilities more effectively and faster.

The remainder of this chapter is organized as follows. Section 2 provides background information on vulnerability assessment, attribution, and exploitation, along with a use case. Section **Error! Reference source not found.** presents the state-of-art vulnerability assessment tools, data sources, and analytics. Section 4 first provides comparison of some security information and event management (SIEM) tools and then presents our temporal and causality analysis to enhance the analysis and management of vulnerabilities, exploitations, and intrusion alerts. Concluding remarks are made in Section 5.

2 Vulnerability Assessment, Attribution, and Exploitation

This section presents basic background information on vulnerability assessment, scoring, and attributes and then discusses a use case on the identification of attribution and exploitation within a cyber analytics environment.

2.1 Vulnerability Assessment

In general, vulnerability refers to any weakness of information technology, assets, or cyber-physical or control systems that could be exploited to launch an attack by adversary. Vulnerability identification, detection, and assessment are essential to cybersecurity, particularly risk assessment. Any combination of security penetration tests and auditing, ethical hacking, and vulnerability scanners may be used to detect vulnerabilities at various processing layers of information, communication, and operations of a system within a cybersecurity environment. Once vulnerabilities are identified, they are ranked with respect to severity and risk score. This helps determine the order in which the prioritized vulnerabilities are put through the patching or recovery process to mitigate system risk, while maintaining system functionality at an acceptable level. To develop a reasonable assessment for a vulnerability, its meaningful attributes should be determined and quantified dynamically by considering system and environmental conditions, as well as its relationship with other relevant vulnerabilities in the space and time domain.

The minimal software attributes of a vulnerability can be listed as authentication, access complexity, and access vector. The minimal impact factors that need to be taken into consideration in case of vulnerability exploitation are confidentiality impact, integrity impact, and availability impact. In general, an attack (e.g., a denial of service attack) can exploit a vulnerability at various network layers, including physical layer (e.g., wireless jamming attack),

MAC layer (e.g., an attack forging address resolution protocol), network and transport layers (e.g., an attack degrading the routing and delivery of information), and application layer (e.g., an attack making intensive requests to overwhelm computer resources). A dynamic accurate assessment of detection capability, exploit likelihood, and exploitation impact associated with a vulnerability assists network defenders and decision makers in improving the assessment of situational awareness and risk of a system. Our approach to achieving such accurate assessment is to determine dynamically not only individual vulnerability attributes and characteristics, but also dependencies, interactions, and probabilistic correlations among vulnerabilities, and then to harness the power of big data analytics to determine correlations and temporal causality among vulnerabilities and cyber events. The vulnerability dependencies and correlations of assets can provide cues about the severity of their attack surface.

Given that zero-day vulnerabilities and exploits always exist, it is critical to have timely detection and control of vulnerabilities and attacks, along with timely recovery and patching of vulnerabilities. For controlling and limiting damage of vulnerability exploitations as well as providing mission assurance, the basic tasks include determining the following: criticality of assets (to a dynamically evolving mission landscape), infection and exploitation status of assets, the movement and propagation paths of exploits, exploitation likelihood, impact and spread of attacks, recognition of adversary strategies and activities, and mission assurance requirements. The common objective of all these tasks at a high level can be expressed as providing real-time detection, containment, and control of vulnerabilities and attacks over a cybersecurity environment that ideally supports at least the following five features: (*i*) use of end-to-end visibility and observability tools across an enterprise network system; (*ii*) understanding the

context and correlation of data, user, and adversary activities; (*iii*) performing real-time analysis; (*iv*) implementing an in-depth defense by monitoring networks and detecting compromised assets and attacker activities; and (*v*) reducing damage and dwell time of attacker within network [1]. The adverse impact of vulnerability exploitations should be minimized by controlling their spread and maintaining mission assurance of systems and operations.

The minimal software attributes of a vulnerability can be listed as authentication, access complexity, and access vector, as stated in Common Vulnerability Scoring System (CVSS) [7-10]. CVSS indicates that the minimal impact factors to consider in case of a vulnerability exploitation are confidentiality impact, integrity impact, and availability impact. Although vulnerability scoring in CVSS and similar type of systems are carefully designed using expert knowledge, they are still inherently ad hoc in nature and possibly assign scores incorrectly to some vulnerabilities. Therefore, it is highly desirable that security evaluation of both individual and collective assets is conducted objectively and systematically [11]. CVSS provides a score for each new software vulnerability discovered that prioritizes the importance of the vulnerability. However, the existing methods and by-default standards such as CVSS do not take into consideration varying conditions in time, environmental factors, and collective behaviors of vulnerabilities and attack impacts, nor does it make unrealistic assumptions about cyber vulnerabilities, exploits, observations, and their models.

The current CVSS base score aggregates several factors: access vector, access complexity, authentication, confidentiality impact, integrity impact, and availability impact. However, it has two main shortcomings. First, only an atomic attack (i.e., a single-stage attack) is considered. Second, as a direct consequence of the first, the damage of assets that would be a result of multi-

stage incremental attacks is not included in the vulnerability assessment. In current CVSS, the base score is a function of access vector, access complexity, authentication, confidentiality impact, integrity impact, and availability impact, where only atomic attack (i.e., single-stage attack) is considered, and no damage on assets is included. In [11], we introduced both theoretical and experimental methods to enhance the assessment of vulnerabilities and vulnerability exploitations, starting initially with CVSS scores and Bayesian network of vulnerability dependencies, and then using Markov models for identifying the most probable exploited vulnerabilities.

One shortfall with vulnerability assessments as they exist today is that the level of criticality of the vulnerability is associated only with the vulnerability itself, but not with the exposure of that vulnerability to an attacker. A technological mechanism that could help address this shortfall is to cross-correlate the existence of a vulnerability with the occurrence of known signatures of adversary behavior. These signatures could be event logs on a system, or specific combinations of event logs that occur within a given timeframe, or they could be based on traffic patterns such as a sudden increase in outbound volume of data. The co-occurrence of the vulnerability with anomalies in system logs or traffic patterns is an indication that the criticality assessment of the vulnerability should be escalated. Furthermore, if the vulnerable host is buried deep inside several layers of security apparatus, it is important to be able to trace the traffic as it crosses through the various proxies and firewalls all the way to an attacker on the Internet, in order to assess the risk to the programs or missions being supported by the vulnerable host.

There are many shared challenges in traffic attribution and in discovering co-occurrence of

vulnerability and system or traffic anomalies. The relevant data are often difficult to identify because the different tiers in the security apparatus collect disparate data from separate locations. Often there is minimal overlap in assets between separate data sets, and even datasets that include shared events and assets but are generated on different hosts can suffer arbitrary timing differences and latencies between associated observations. Network address translation further complicates valid cross-correlation by obfuscating the true start and endpoints of flow records.

Surmounting these challenges to improve vulnerability assessment requires a centralized data store, coupled with a process that aggregates data streams from multiple sensors, normalizes the data across the different sources to allow pivoting from data collected in one location to data collected in another, and labels the data with appropriate knowledge engineering to provide analysts ready access to the data coupled with discoverable knowledge about the provenance and contents of the data. Centralization of different data streams would enable automatable analytics to simultaneously process data collected in multiple locations.

Developing improved vulnerability assessment apparatus is likely to be an iterative process in which an analyst explores various correlations and patterns in the data, forms a hypothesis, tests the hypothesis by querying the data, develops a more robust signature for the attack mode under investigation, and automates the association of that attack mode with a known vulnerability using the signature and the data available in the data store. An example of a portion of this process, leveraging the Scalable Cyber Analytic Processing Environment (SCAPE) technology [6], is carried out at the U.S. Army Research Laboratory.

2.2 Use Case: Identification and Attribution of Vulnerability Exploitation

A computer network defense service provider (CNDSP) is an accredited organization responsible

for delivering protection, detection, response, and sustainment services to its subscribers [30]. Such an organization typically assembles large datasets consisting of IDS alerts, firewall logs, reconnaissance scans, network traffic patterns, and other computer monitoring data. In this particular example, such CNDSP-collected data have been stored in an Accumulo database, which has been made available to an analyst for data exploration purposes via the SCAPE (formerly known as LLCySA [6]). This is illustrated as a big data cyber analytic system architecture in **Figure 1**.

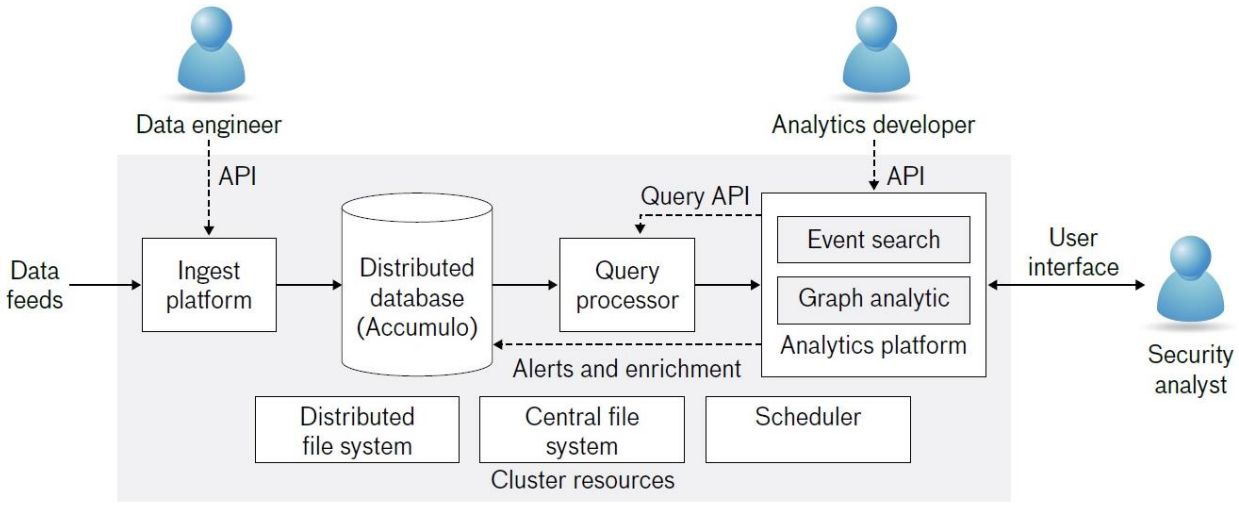

Figure 1: A big data cyber analytics framework [6].

The SCAPE environment provides knowledge engineering that allows an analyst to access the data without detailed a priori technical expertise regarding where data have been collected, which sensors have been deployed, or knowledge of the data storage format and schema. In this particular example, the goal is to identify an attack deep inside the DoD network and trace the net flows back to an attacker on the Internet. Host intrusion data are used to provide the initial tip. SCAPE is used to conduct an interactive investigation, pivoting between different relevant data sources to develop a hypothesis and confirm illicit activity. A simple aggregating analytic

identifies a subset of hosts with the highest number of Host Intrusion Protection System (HIPS) alerts. Using SCAPE, the analyst pivots to the associated NetFlow data communicating with these hosts and identifies a suspicious flow revealing a late-night surge of Server Message Block (SMB) activity for one of these Internet Protocols (IPs). This process is depicted on the right hand side of **Figure 2**, which also shows a plot of NetFlow activity associated with this time period. Comparison with the previous several days of traffic suggests that the volume of data exchange on December 11th is potentially atypical for the host in question.

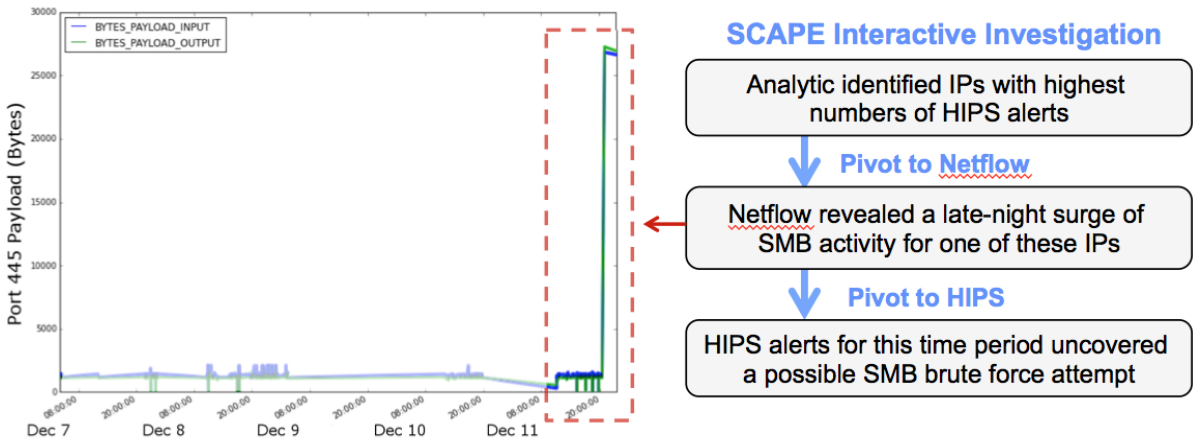

Figure 2: Sample SCAPE workflow and resulting graphic output.

The SCAPE environment provides an easy-access interface to multiple cyber data sources, allowing an analyst to quickly pivot from host intrusion protection events to NetFlow. The correlation of HIPS alerts with suspicious flow activity may imply that the assessment of the associated vulnerability should be escalated to a higher level of priority.

Closer inspection of the HIPS data on the host uncovered evidence in this time period of a possible SMB brute force attempt. The next step in this investigation would be to aggregate

network address translation logs from the various firewall and proxy devices that intervene between this host and the remote host on the open internet. Doing so would allow the analyst to determine the destination of the large outflux of data.

An analysis of this type could be used to improve an existing assessment of the vulnerability associated with this host, and others like it. Having discovered the specific signatures associated with a breach of this type, the co-occurrence of the existing vulnerability with significant HIPS activity or with significant changes in network traffic could be used to escalate the associated severity of the vulnerability, indicating that a higher priority should be designated to this host, because of indications of potential exposure to an adversarial entity.

3 State-of-Art Vulnerability Assessment Tools, Data Sources, and Analytics

3.1 Vulnerability Assessment Tools

Vulnerability is thought to be the intersection of three elements: a system susceptibility or flaw, attacker access to the flaw, and attacker capability to exploit the flaw [3]. Many vulnerability assessment tools exist, both in industry and within the TNUB, to detect the presence of such flaws. The tools typically leverage extensive databases of known software vulnerabilities and itemize the observed malware and other attacks that leverage each vulnerability to assess its severity. Network-based scanners perform credentialed or uncredentialed scans of endpoint hosts to enumerate open ports, identify which software is installed, and detect missing patches. Web application and database scanners check for flaws in data validation and other mechanisms for command injection or information leakage. Host-based scanners look for known problems, such as viruses, or faulty operating system configurations to identify security gaps. Collectively, these tools exhibit a weakness; the scan can identify a system flaw or susceptibility, and a database can

estimate the attacker capability to exploit the flaw, but none of the tools is equipped to quantify the extent to which an attacker can access a flaw. The fundamental reason driving this weakness is that all current vulnerability assessment processes are inherently local to each host. Assessing attacker access to a flaw is inherently a non-local problem that involves not only the vulnerabilities on a given system, but also the vulnerabilities of systems that are connected to it on the network.

3.2 Data Sources, Assessment, and Parsing Methods

Identifying data sources for use in vulnerability assessment and exploitation is a straightforward proposition. There are literally hundreds, if not thousands, of security tools and information technology systems that generate data useful for enhancing or enriching an organization's situational awareness posture and providing content pertinent to a vulnerability assessment and exploitation exercise. However, the challenge is not in finding the data sources but rather adopting approaches or tools that aggregate and correlate the data in a meaningful manner.

To illustrate this point, let us walk through a hypothetical data collection exercise in preparation for a vulnerability assessment. For the sake of simplicity, consider three data sources in this example, although there could be dozens of data sources in an actual assessment. The first data source is Nessus, which is an industry-recognized vulnerability assessment scanning tool. Nessus is developed and maintained commercially by Tenable Network Security and provides a variety of features, including the following: network vulnerability scanning, application vulnerability scanning, device compliance assessment, and network host discovery.[1] The second data source is McAfee ePolicy Orchestrator (ePO), which is an industry-recognized, host-based security tool. The ePO is developed and maintained commercially by Intel Security and provides many

features, including the following: host intrusion prevention, policy auditing, and anti-malware.[2] The third data source is Snort,[3] which is an open-source network intrusion detection and intrusion prevention tool. Martin Roesch developed Snort for public use and dissemination in 1998. Snort is freely available for download, and there are hundreds of thousands of community members that use, maintain, and contribute to the tool. Snort provides intrusion detection and prevention capabilities by means of network traffic analysis and packet capture.

These example tools all independently provide some degree of situational awareness. A naive approach toward vulnerability assessment would be to consider the output from each tool in isolation. So, Nessus output would be used for application vulnerability assessment, McAfee ePO would be used for host policy compliance assessment, and Snort would be used for exploitation detection and assessment. Although this approach may be straightforward to understand and easy to implement, there is no correlation occurring between the different data sources, leaving the potential for major holes in the analysis of vulnerability exploitation process.

As an illustrative example, imagine a scenario where a Snort subject matter expert named Samantha is providing intrusion detection analysis services to a small organization. During her shift, Samantha receives two alerts identifying unauthorized remote access attempts on two separate network segments: Alpha and Beta. Without consulting any additional information sources, how would Samantha assess which alert to investigate first? The alerts are identical (i.e., triggered by the same intrusion detection signature), so there is no clear way to gauge which subnet should be prioritized. Samantha could assess the alerts in a sequential manner based upon

---

[1] http://www.tenable.com/products/nessus-vulnerability-scanner
[2] http://www.mcafee.com/us/products/epolicy-orchestrator.aspx

the time of notification and investigate subnet Alpha first. However, what if the software patches for the assets in subnet Alpha are all current, whereas in subnet Beta they are months old? Snort cannot detect this, but a vulnerability scanner such as Nessus can. Moreover, what if the assets in subnet Beta have not received updated anti-virus signatures in weeks, but subnet Alpha received the latest definitions the previous night. Again, Snort does not have visibility, but a host-based security system such as McAfee ePO does. Furthermore, what if the alerts have a causal relationship? Insights provided by other tools could establish such relationships and provide the analyst with means to detect future attacks. The failure to establish even the most basic of associations between the various tools and data sources ultimately increases the risk of more vulnerability exploitations on Samantha's watch.

An improved approach is the analyst-to-analyst or ad hoc correlation of tools and sources. Some refer to this as the "swivel-chair" approach because it involves an analyst turning in her chair in order to request assistance from a colleague operating a different tool. The swivel-chair approach mitigates some of the concerns in our hypothetical scenario by Samantha engaging with her team of colleagues. The swivel-chair approach is an improvement over relying on a tool in isolation because diverse data increase the probability of making better-informed decisions. However, this approach suffers from its own drawbacks: namely, timeliness of information gathered and consistency of analysis. What questions will Samantha ask of her colleagues? Will her colleagues interpret her questions correctly? Will her colleagues be able to provide her with relevant responses within the same temporal domain as the alerts she is investigating? How long will it take for Samantha to receive responses from her colleagues? More importantly, if Samantha and her colleagues miss something critical, there is no digital record of the swivel-

---

[3] https://www.snort.org/

chair exchange, and no way to track which observations led the team to the direction they ultimately took in the investigation. Human fatigue may also play a role in increasing the risk of errors. In addition, different levels of education and experience will yield different analysis methodologies. As a result, manual or ad hoc correlation of data sources is also problematic and may not yield consistent and comprehensive results.

A more formal and analytic approach to vulnerability analysis may improve reliability and produce actionable results. This approach stages the various data sets to support a variety of interfaces and visual representations of the data. This approach also ensures that relationships established between the various data sets are stable with consistent and unique key values. The approach is also the basis for data analytics. Data analytics includes a conceptual data modeling process that needs to be applied to the various data sources. This process helps the understanding of underlying attributes of the individual datasets and the current schema of the individual datasets. The common attributes across the datasets serve to establish relationships between the data sources. In our example, all three data sources share IP address information. When modeling these data, keying on the IP address would be one method to allow for a comparison of the elements across the data sources. In addition, during the data modeling process, a common taxonomy or data dictionary for the data elements of interest should be established. The data dictionary is an important tool to establish proper relationships between entities in the different data sources. All three data sources have multiple references to IP addresses in their schema. McAfee ePO references IP in multiple ways, including the following: AnalyzerIPv4, SourceIPv4, TargetIPv4, and IPAddress. Nessus has several references, including the following: IPS, Host IP, Scanner IP, and IP. Snort also has several references, including the following: IPv4, IP Source, and IP Destination. Without a taxonomy defining the various IP elements across

the data sources, establishing a relationship using McAfee SourceIPv4 and Snort IPv4 may yield incorrect results. Indeed, even though they are both IPv4 addresses, they do not necessarily represent the same node.

Once the data modeling process is complete, the data elements identified in the modeling phase need to be extracted and stored in a common data format. This phase consists of developing parser(s) to extract, transform and load (ETL) the data. Multiple parsers may be required for each data source in order to account for different input formats. In our example, each of our tools has various output formats, including the following: XML, JSON, CSV, CEF, and PCAP. The parsing process involves extracting the data attributes of interest, tagging the attributes with metadata and taxonomic details, and outputting the data in a common format for efficient querying. In addition, unlike the swivel-chair approach, the analytics approach automates the majority of the steps after modeling the data. Automation ensures a consistent stream of correlated data is available to support a vulnerability event and affords a decision maker more time to take an appropriate course of action. In our example, all three tools have Application Programming Interfaces (APIs) that allow for the programmatic extraction of data, in order to be used in other applications. Automating this approach would be as straightforward as reviewing the respective API documentation for each tool and writing scripts in order to extract the attributes of interest. The APIs generally support high-level programming languages (i.e., Python, Java, Perl, etc.). In addition, many of these software manufacturers and support communities already have preconfigured scripts that can be tweaked to fit most purposes.

4 Secure Management of Cyber Events Involved with Vulnerability and Exploitation

This section first describes the basics and comparison of three well-known SIEM tools. Then, to

enhance the dynamic analysis and management of cyber events, we present the basic idea and method behind temporal causality analysis by addressing the Structured Query Language (SQL) injection attack.

4.1 Comparison of Current SIEM Tools

SIEM tools are designed to correlate a variety of log events in order to enhance an organization's situational awareness. SIEM tools accomplish this by collecting log events from multiple data sources and across numerous hosts, leveraging a variety of analytical methods to establish relationships between disparate events, and, finally, providing security analysts a central console for managing and visualizing events in a unified manner.

Each SIEM product has features that set it apart from competing products; however, at a minimum, each SIEM must provide three basic capabilities: namely, a mechanism for data ingestion, a mechanism for event correlation/analysis, and a mechanism for reporting and visualization. Many SIEMs go further and offer additional capabilities such as native integration with common log and event generating tools, providing threat intelligence feeds, enhanced logging via deployable agents, and automatable response capabilities. SIEM tools are a critical part of every industry due to the sheer volume and velocity of events that organizations generate on a daily basis. It is literally impossible for a security analyst to review pages upon pages of log events from dozens of sources and expect to pinpoint threats and vulnerabilities with any reasonable degree of accuracy or timeliness. The usage of SIEM tools is not exclusive to activities in the vulnerability assessment or cybersecurity domain; however, these activities greatly benefit from the use of SIEMs for two major reasons:

i) First, SIEMs can detect anomalous events obfuscated by large volumes of benign traffic. By correlating events from a variety of sources, it becomes increasingly difficult for an attacker to hide actions that would not occur during "normal" business operations. Individual events in operating system logs, application logs, firewall logs, and directory service logs may seem innocuous, but through the lens of a SIEM, relationships that were once invisible become transparent. For example, take the following four events: (*i*) user downloads an email attachment on her workstation, (*ii*) workstation makes Domain Name System (DNS) requests to several unknown domains, (*iii*) workstation attempts the installation of an unsigned executable, (*iv*) workstation experiences a spike in outbound network traffic over Transition Control Protocol (TCP) port 443. Collected from different logging systems and assessed independently, these events may or may not raise red flags. However, upon correlation and investigation as one unified event in multiple stages, this activity could be deemed highly anomalous, potentially malicious, and warrant further investigation.

ii) Second, SIEMs can enhance the efficacy of incident-handling practices. By automating the correlation and aggregation of cyber events, providing reports and descriptive statistics, and, in some instances, supporting automated responses, SIEMs can be thought of as a virtual incident response team that helps a security analyst prioritize what is noise or benign and what is suspicious or malicious. SIEMs use a variety of statistical and analytical methods to transform and relate events over long periods of time. For example, a low and slow data exfiltration event is difficult to detect because it occurs over an extended period of time (i.e., weeks, months, or longer), and only a

small fraction of a target file is transferred during each session. A web log entry documenting a 10 MB HTTP transfer to a public web server on any given day is uninteresting and quite common. On the other hand, a SIEM that correlates six months' worth of web logs and discovers a 10 MB HTTP transfer each day to the same web server is quite interesting and suspicious!

In this section, we discuss three SIEM tools: one open source, one traditional, and one non-traditional. We broadly focus on their strengths and weaknesses in five major areas: cost of adoption, correlation capabilities, compatibility with common logging sources, threat intelligence capabilities, and scalability. We did not include visualization capabilities as one of the criteria of comparison because the topic is highly subjective.

Figure **3** below summarizes of our findings.

	Correlation Capabilities	**Compatibility with Common Log Formats**	**Threat Intelligence**	**Scalability**
OSSIM	Handful of community submitted correlating rules. Custom engine to create new and more complex rules.	Native support for a variety of logging formats including server logs, vulnerability assessment tools, and system monitoring tools	AlienVault Open Threat Exchange. Community of interest feeds, free and open source	Limited to deployment on a single server
HP ArcSight ESM	HP proprietary CORR-Engine for optimized log and event correlation. Hundreds of out-of-the-box rules configured for perimeter and network security monitoring	Smart connectors and flex connectors used to parse raw log/event data into ArcSight's common event format (CEF). Forwarding connectors used to export events from ESM to other tools in CEF format.	STIX/TAXII compliant threat intel providers, such as Verisign iDefense (requires a separate subscription) or HP Threat Central	Highly scalable. Vertical scaling unnecessary supports clustered deployments.
Splunk	Search Processing Language (SPL) supports statistical and analytical correlation; manual correlation across universally indexed data based on temporal domain or field values. Automatic	Handles raw log files with universal indexing. Automatically separates log stream into searchable events. Supports manual indexing for custom feeds.	Enterprise Security App for ingesting external threat feeds and correlating the indicators of compromise with existing events indexed in Splunk.	Highly scalable. Vertical scaling unnecessary supports clustered deployments.

	correlation of events with similar field values.			

Figure 3: Summary of SIEM tool comparison.

4.1.1 Open Source SIEM tools

AlienVault Open Source Security Information and Event Management (OSSIM) is a community supported open source SIEM tool and the "lite" version of AlienVault's commercial SIEM: Unified Security Management (USM) [12]. The OSSIM project began in 2003 in Madrid, Spain, and it became the basis of the Alien Vault Company founded in 2007. Of the three tools presented in this chapter, OSSIM is the only one that is free to download and use without restriction. However, the cost of adoption is not free. Using and supporting OSSIM requires time and effort in reviewing documentation, posting questions on online forums, and researching functionality. OSSIM has an online threat intelligence portal called the Open Threat Exchange (OTX) that gathers daily threat events with indicators of compromise, referred to as pulses [14]. The portal is configured in a publish-subscribe fashion, so anyone in the community can publish pulses, and anyone in the community can subscribe to specific publishers and feeds of interest. Threat intelligence is an important but often understated aspect of the vulnerability assessment process. It is a critical component because these indicators act as a supplement for threats, and an organization should be on the watch for these indicators. For example, most threat intelligence services (including OTX) host a list of known bad IPs. This list can be used by a SIEM as a watch list, and any traffic originating or destined for one of these bad domains should immediately be flagged as suspicious.

OSSIM can natively parse and ingest a variety of common logging sources, including the following: Apache, IIS, OpenVAS, OSSEC, Nagios, Nessus, NMAP, Ntop, Snare, Snort, and Syslog. OSSIM leverages regular expressions to parse data, which allows for a custom parser to

be written and extend its support to any data source that outputs in a text format. In addition, OSSIM comes equipped with a host IDS that can be deployed as an agent for collecting system and log events if no preferred collection tools are present in a given environment [12].

OSSIM performs correlation by relating events in a sequential and temporal fashion. OSSIM comes packaged with a handful of built-in directives for common cyber events such as brute-force attacks, DOS attacks, enumeration, and fingerprinting scans, etc. Beyond the handful of preconfigured templates, OSSIMs have a correlation engine that allows for the creation of custom directives. **Figure 4** below illustrates a logical sequence of events that could be built as a custom correlation directive in OSSIM to uncover a potential brute-force attack [13]. At each level, an alert can be sent to the appropriate parties indicating a potential threat. Furthermore, time can be introduced as an additional attribute of interest, so that if the failed logins are occurring at a particular frequency, a different alert or severity level can be triggered.

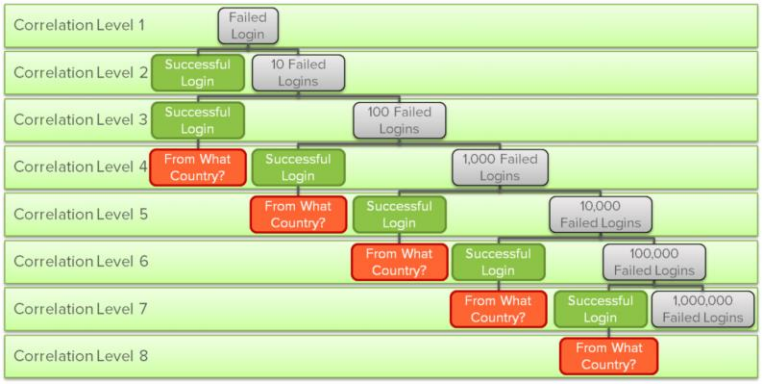

Figure 4: Example brute-force attack correlation logic.

OSSIM is not an enterprise class SIEM and cannot scale beyond one host or server. If there is a need for a large deployment, OSSIM's commercial counterpart USM provides support for horizontal scaling (adding more servers instead of buying bigger servers) and may be a better choice. OSSIM is the least expensive USM license [12] that includes hundreds of professionally developed correlation directives and forensic logging capabilities not included in the open source

version. OSSIM does not support integration with big data technologies such as Hadoop, nor does it natively support exporting of events to external relational databases. OSSIM supports basic authentication or integration into directory services such as LDAP or Active Directory. More information about OSSIM and a free download of their SIEM software (ISO format) can be found on their website [32].

4.1.2 Traditional SIEM tool

ArcSight has been developing SIEM tools since 2000 and is one of the oldest players in the market. In 2010, Hewlett Packard (HP) acquired ArcSight USD [28] and extended their portfolio of services to include enterprise cybersecurity. Today, HP ArcSight Enterprise Security Management (ESM) is arguably the most heavily adopted SIEM tool by commercial and government organizations alike. ESM takes a modular approach towards SIEM. The standalone configuration of ESM excels in the three basic requirements of a SIEM tool (i.e., ingest, correlation, and visualization). Additional features such as central log management and threat intelligence can subscribed to and deployed separately, to further enhance the capabilities of ESM. The cost of the ESM software and professional support is not clear, and it appears to fluctuate based upon the deployment configuration, number of data sources, and volume of ingest.

ESM has hundreds of built-in features all configurable from the ESM graphical user interface. If configured properly, these built-in features can substantially increase the resolution of an incident handler and decrease her time of response. Some of the features include the following: (*i*) data enrichment with user, asset, or key-terrain information; (*ii*) prioritization and normalization of events; (*iii*) “near-real time” correlation of data and threat intelligence; (*iv*) data forensics and historical and trending analysis; and (*v*) a vast library of predefined security use

cases, compliance automation, and reporting tools, which are designed to minimize time spent on creating compliance content and custom reports; and (*vi*) workflow automation, which generates alerts and escalates events based on elapsed time [15].

ESM is feature rich and, as a result, there is a steep learning curve. As seen from **Figure 5** below, the ESM console is loaded with options, and it is arguably the least user-friendly interface. ESM protects its user console and data with authentication and authorization via directory service (e.g., LDAP, Active Directory) integration and role-based access control.

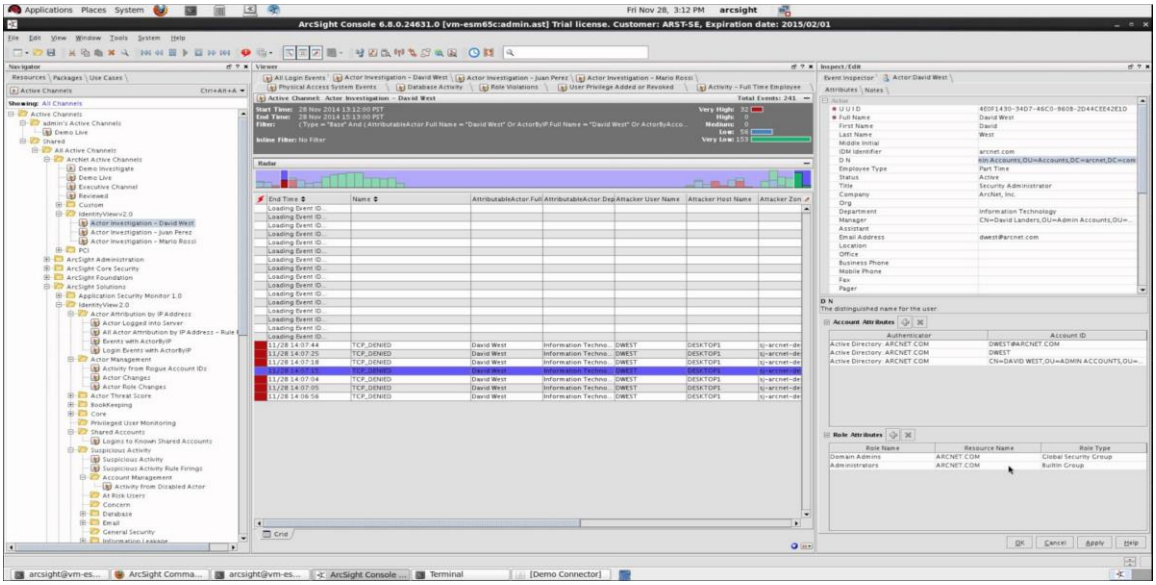

Figure 5: ArcSight ESM 6.8 console.

For threat intelligence, ESM takes a standards-based approach and can receive feeds in the STIX or TAXII formats. STIX stands for Structured Threat Information eXpression and TAXII stands for Trusted Automated eXchange of Indicator Information. Both standards are part of cybersecurity information sharing efforts led by the Department of Homeland Security. A list of threat intelligence providers that support STIX and TAXII can be found at https://stixproject.github.io/supporters/. In addition, HP provides a community intelligence portal "HP Threat Central" that hosts private security forums, threat databases, and anonymized

indicators of compromise (IOC) [19].

Out-of-the-box, the ESM smart-connector natively supports the parsing, ingestion, and conversion of hundreds of industry-recognized technologies into ArcSight's Common Event Format standard. Some technology examples include the following: operating systems (Microsoft, Apple, Redhat Enterprise Linux, Oracle Solaris) anti-malware tools (Kaspersky, McAfee, Symantec, Trend Micro), application security (Bit-9, RSA, McAfee), network devices (Cisco, Juniper), and cloud (Amazon Web Service) [18].

In the event that the smart-connector does not support a particular feed, a custom feed can be written using the ESM flex-connector. The flex-connector framework is a software development kit (SDK) that enables the creation of a smart-connector tailored to the specific event data format [16].

ESM performs log correlation via the proprietary HP ArcSight's Correlation Optimized Retention and Retrieval (CORR) Engine. The CORR engine is a flat file system optimized for read performance. According to the ArcSight team, it is 5× more efficient at event correlation and 10× more efficient at data storage [17] than the previous SQL-based correlation engine.

ESM has rule-based, statistical, or algorithmic correlation, as well as other methods that include relating different events to each other and events to contextual data. In addition, ESM has hundreds of preconfigured rules for advanced correlation and with the integration of threat intelligence sources; the correlation engine can quickly identify IOC [18]. A few examples rules that are available out-of-the-box include the following: top attackers and internal targets, top infected systems, top alert sources and destinations, bandwidth usage trends, and login activity trends [19]. More information about HP ArcSight ESM can be found on their website [33].

4.1.3 Non-traditional SIEM tool

In 2002, Eric Swan and Rob Das [24] founded Splunk on the premise that it would serve as the Google (i.e., search engine) for enterprise log data. Splunk satisfies the basic requirements of a SIEM (i.e., data ingest, correlation, and visualization), but it is not a traditional SIEM. Traditional SIEMs often have a fixed schema of attributes that can be correlated against one another. Data are ingested and bucketed into those attributes and then correlation rules are applied to establish relationships and insights. Splunk was designed in a more flexible manner to ingest any type of log data, automatically index it, and extract searchable events. If Splunk's automatic indexing is off base, manual user intervention can be taken to tweak the indexing for a specific data source/type. Once the data are indexed and searchable, Splunk offers a variety of methods to interact with the data, including methods that support enterprise security use cases. Splunk initially offers its product for free via a 500MB/day data-indexing license. However, 500MB/day can quickly be consumed in minutes when multiple sources are being indexed. Splunk's cost model is based on the volume of raw and uncompressed data indexed per day. Similar to ESM and OSSIM, Splunk protects its user console and data stores with authentication and authorization via directory service integration and role-based access control.

Splunk includes an application for enterprise security [25] that supports ingestion of external threat feeds for correlation with log events. Splunk itself does not offer any threat intelligence feeds or host a threat intelligence portal. It relies on external feeds and can support both open source as well as subscription-based models. External threat intelligence feeds can be thought of as an additional data source that Splunk can automatically ingest, index, and create searchable events.

Splunk does not rely on predefined correlation rules. Splunk's approach toward event correlation

is to provide the end user with a powerful search processing language (SPL) and present her with a unified, indexed, and searchable database of events. SPL is designed to transform statistical correlation methods into queries across the unified search database [26]. This pool of indexed data can also be searched in a manual fashion, and events can be correlated on the basis of time of occurrence or attribute of interest. Splunk can also perform automatic correlations based upon event attributes with similar values. An added advantage Splunk has over some of its competition is that if an export is available, Splunk can ingest and correlate events processed by other SIEM tools.

Splunk is highly scalable as illustrated in **Figure 6** [27]. Depending upon the use case, all of the Splunk roles can run on a single host or run distributed across hundreds of hosts.

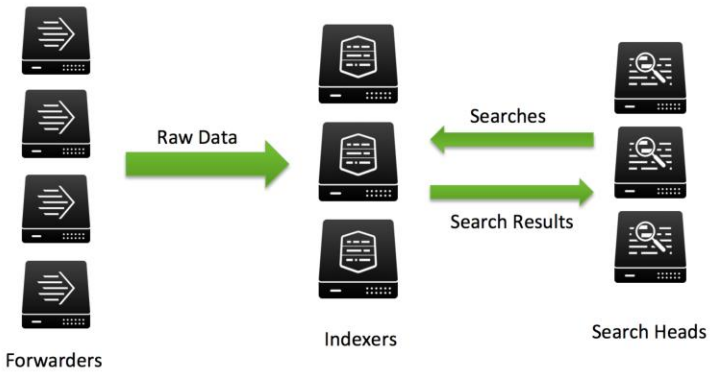

Figure 6: Splunk three-tier architecture.

More information about Splunk and a 60-day free trial download of the Enterprise version of their software can be found on their website [34].

There are various documented reports that quantify the return on investment that SIEMs tools provide. HP commissioned one such investigation where they identified millions in time and monetary savings across a variety of organizations [28]. The majority of the savings appear to be the result of SIEMs tools transforming assessment work from a large team of analysts to a

handful of specialists. For many industries, SIEM tools are the current answer to the question of how to enhance situational awareness and look at vulnerabilities from a holistic perspective. Unfortunately, many SIEM tools, including the three discussed in this chapter, suffer from drawbacks, resulting in a false sense of security.

The majority of SIEM tools such as OSSIM and ArcSight ESM rely heavily on rule-based correlation. As a result, these systems require frequent tuning in order to account for false positives and false negatives. As the volume of data increases, so do the false positives. As correlation rules are tuned to account for false positives, the number of false negatives increases. This is a known and often poorly addressed fact of SIEM and standalone security tools alike. Splunk's approach is a bit better than the traditional SIEM tool because it doesn't focus on pre-defined correlation rules. Giving the end user a language (e.g., SPL) with which to interact with the data is a step in the right direction. However, this approach is limited by how creative an analyst's queries are and how well the language can transform an analyst inquiry into queries across the data.

SIEM tools cook the data (e.g., pre-processing and normalizing) using a variety of methods that primarily support their pre-defined correlation capabilities. This approach works well to detect and mitigate known threats, but unknown threats are still a problem. Again, Splunk's approach is somewhat better in that it attempts to automatically index and make all the data searchable. The question is, however, how accurate is their automatic indexing? It is also convenient that Splunk's cost model is based upon volume of indexing.

Horizontal scaling for SIEM tools only slightly mitigates bandwidth and storage constraints. If there is a spike in volume or velocity of log data for a given organization, adding more nodes to enhance a SIEM tool's capacity and performance only works until the next spike occurs, not to

mention the challenges of using SIEM tools in a decentralized model. For instance, if an enterprise is geographically dispersed between two continents, how will the logs from one site be transferred to the other site in order to be processed by a SIEM tool in a timely fashion? The answer is they will not be transferred. Each site will likely have its own SIEM tool infrastructure on premises with some mechanism to cross-correlate the data. This is not a trivial proposition. Some SIEM tool manufacturers have started offering "cloud"-based models to better support this use case, but it is not yet clear whether this approach is beneficial.

4.2 Temporal Causality Analysis for Enhancing Management of Cyber Events

This subsection introduces a novel temporal causality analysis for cyber events classified into five processes: namely, attacker, vulnerability detection and protection, intrusion detection, agility, and risk assessment. This temporal causality analysis differs from the current SIEM tools in that it provides vector-time, vulnerability-centric causality pairing graphs, and context-specific vulnerability-centric causality pairing graphs of events including agility and risk actions, which can also provide cues for the detection of zero-day vulnerabilities and attacks. With the help of timestamps of events, the vector-time concept that is imported from distributed systems [31] allows analysts to investigate the events in temporal domain, even if time synchronization is not available among the hosts of a cybersecurity environment. In addition, this causality analysis can incorporate human factor from the perspectives of user, defender, and adversary, although it is not included in this section due to space constraints.

To protect against malware detection and spread control are essential to maintaining the functionality or mission assurance of a system. The success of the protective measures depends on a number of factors, including the accuracy of IDS, the system's resilience against attacks, the

strength of vulnerability patching and recovery, the level of situational awareness, and the correlation of sensor observations and measurements. It is highly desirable to perform real-time data analytics of cyber events, observations, and sensor measurements to discover interactions and characteristics of cyber events. In stealthy malware, the adversary aims to make the malware invisible and undetected to a cyber-defensive mechanism over a target network. To achieve this, the adversary gathers information on the state of defensive mechanisms. In addition, the adversary may choose to obfuscate the real intent by performing misleading activities and operations.

The causal interpretation of networks is essential to understand how events and entities trigger each other, thereby indicating their causalities. Causal models help determine how the sequence of events or entities trigger each other. There can be numerous latent variables within a system that are not observable. Although it may be tolerable to not model some latent variables in answering probabilistic queries, it is highly desirable for causality analysis to identify latent variables when correlations between them represent causal relationships. In general, correlations between any set of variables may form a set of both causal and non-causal relationships. A causal model can be represented as a directed acyclic graph over random variables, where the value of each random variable is computed as a stochastic function of the values of its parents [2]. In [4], the triggering relations in causality reasoning about network events are addressed by comparing rule- and learning-based methods on network data of stealthy malware activities.

Our overall goal is not only to detect vulnerabilities and exploit but also to mitigate the adverse impact of vulnerability exploitations. Indeed, it is highly desirable that the adverse impact of vulnerability exploitations do not lead to an unacceptable level in mission assurance. This may

be achieved by using the approaches of both reactive mitigations and proactive mitigations. Therefore, in addition to considering the cyber events of attacker, vulnerability detection/protection, and intrusion detection, we also consider the cyber events of agility and risk assessment. So, we consider five cyber processes in the temporal causality analysis of cyber events, where an event denotes any observable occurrence of an activity or action in a system or network. An event may have physical attributes (e.g., network topology, frequency of event occurrences over a period), meta-data attributes (e.g., IP addresses, port addresses, timestamps, control or user data types, TCP, or UDP), event interaction attributes (e.g., vector time, where the sequence of past values of vector-time indicate the interaction of causal events of different processes; lag time of event responses), or cross-layer attributes of OSI model (e.g., application type, file type, protocol type). In the figures below, directed edges between events of different processes indicate causality, whereas undirected edges indicate that events exhibit temporal order but are not necessarily causal. To keep track of interactions between five cyber processes, we use the vector-time concept in distributed system events. Vector time characterizes causality and temporal order such that the *k*th entry of a vector time that corresponds to process P_k is incremented by 1 each time an event occurs in process P_k. Whenever an event of P_k receives the vector time of another process' event, the vector time entries of P_k's event are aggregated with the vector time entries of the other process.

As an example, let us consider a SQL injection attack that takes advantage of the ability of influencing SQL queries formed and submitted to a backend database by an application such as a web application using inputs received from potentially untrusted sources [12]. ***Figure 2*** shows that the attacker activities can be classified into at least seven categories, labeled as a_1 to a_7,

corresponding to (a_1) performing reconnaissance, (a_2) exploiting a vulnerabilities of (a_3) the webserver and (a_4) the database server, (a_5) delivering malware to escalate privileges, (a_6) installing a backdoor on system, and (a_7) stealing data. Some of these attacker events involved events of other cyber processes. For instance, network-based IDS and/or host-based IDS may detect some of these attacker events and generate alerts; some vulnerabilities may get exploited and then recovered; agility events may help avoid or mitigate impact of attacks, with the help of risk assessment events; and the tasks prioritization of vulnerability and intrusion detection processes can be strengthened with the guidance of risk assessment events. The causality between different processes in Figure 7 is illustrated by directed edges.

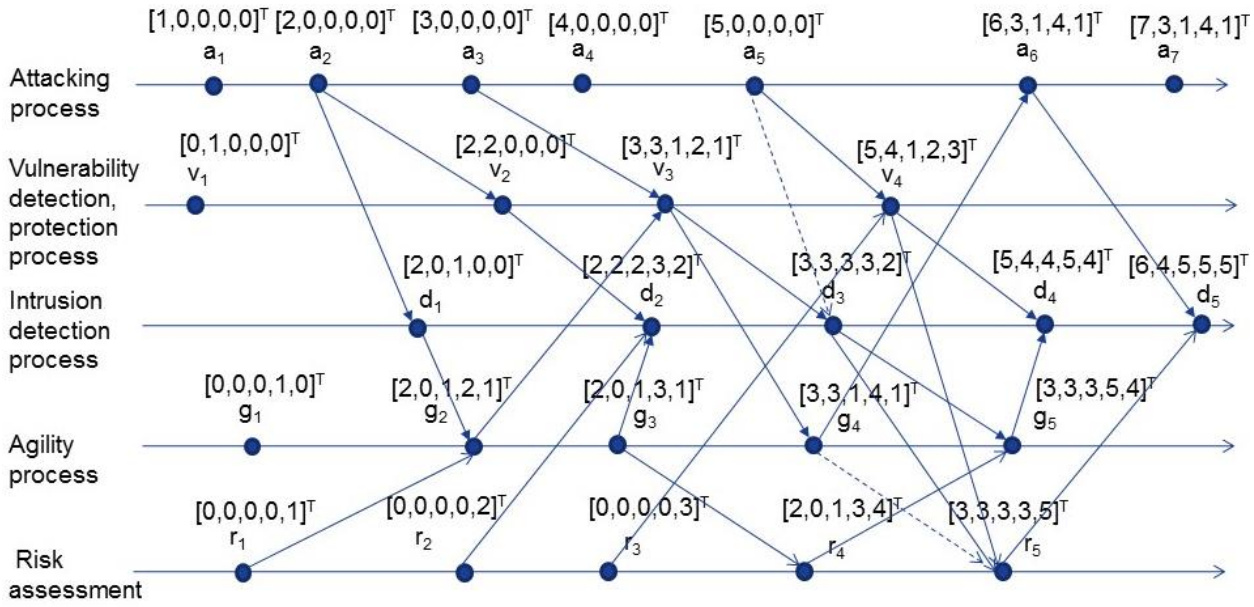

Figure 2: Causality edges of cyber events for a SQL use case. Attacking process events (a_1: reconnaissance; a_2: use SQL injection to exploit v_2 of webserver; a_3: use SQL injection to exploit v_3 of database server; a_4: deliver malware and tools to escalate privileges; a_5: install a backdoor on system by exploiting v_4; a_6: exfiltration of system credentials; a_7: theft of data); vulnerability process events (v_1, v_2: web server; v_3: database server; v_4: backdoor); intrusion detection process events (d_1, d_2, d_3, d_4, d_5); agility events: g_1, g_2, g_3, g_4, g_5); and risk assessment events (r_1, r_2, r_3, r_4, r_5).

Once the causality edges and the vector-times of events are established as shown in Figure 3, time intervals with predefined durations can be designated so that all causal and temporal edges of each time interval can be studied in-depth. **Figure 3** illustrates how all those directed edges that are involved with vulnerability v_4 can form causal pairs. However, some temporal edges can also be causal, and, therefore, the next step is to find out which temporal edges are causal (see **Figure 4**). Then, all causal edges are used to form the so-called vulnerability-centric pairing graph (VCP), as shown in **Figure 4**. The cyber data corresponding to the interactions of the VCP edges can represent the quality data of its time interval. These quality data are stored properly in

the database so that they can be extracted easily and instantaneously by database queries formed by cyber analysts. Hence, big data of cyber events can be reduced to a smaller size of the aggregated quality data of temporal causal events.

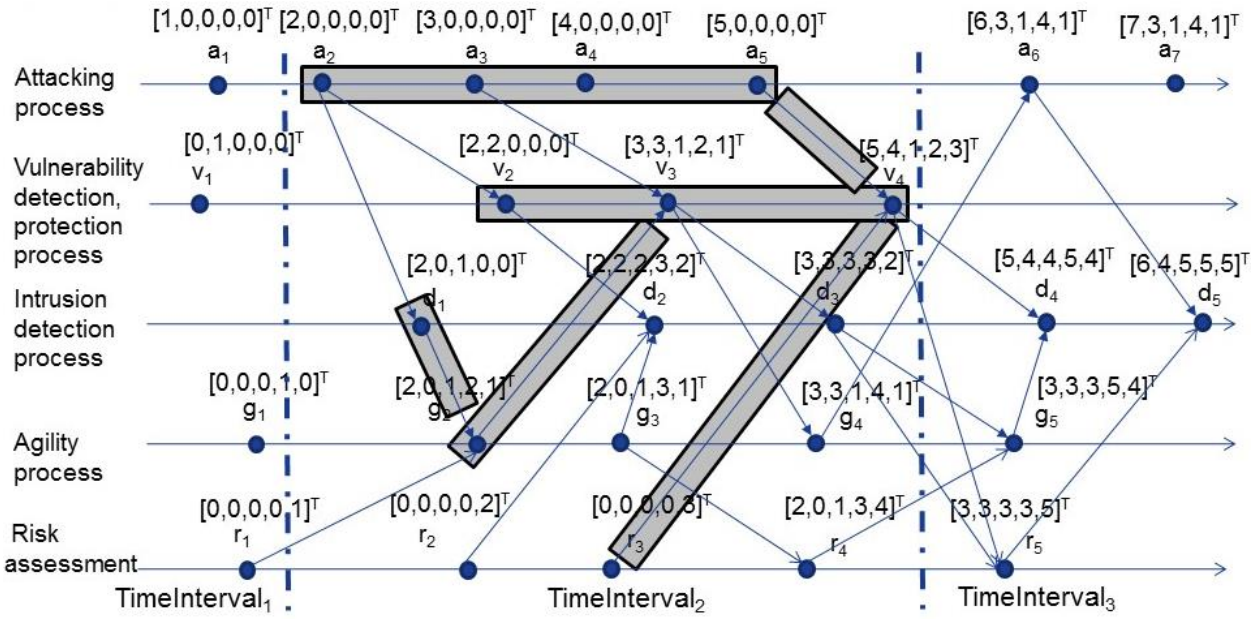

Figure 3: Causality edges of SQL cyber events within the middle time interval, where vulnerability v_4 is found to be exploited.

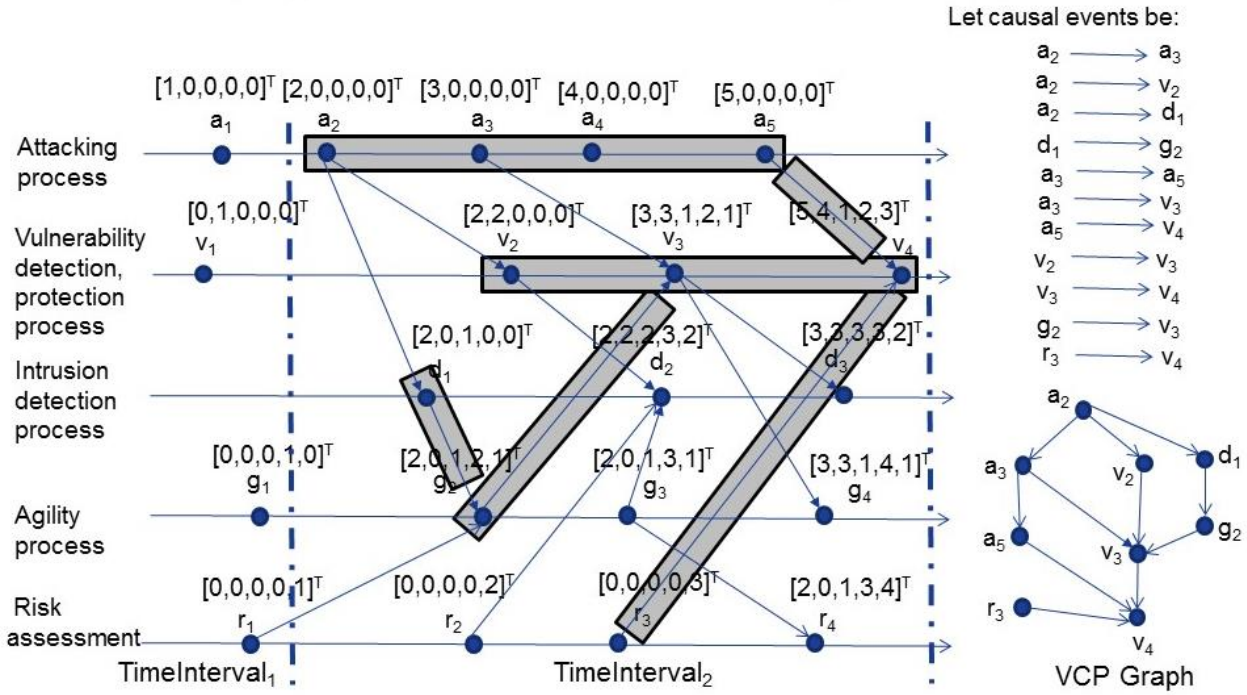

Figure 4: Causality edges of SQL cyber events with the VCP graph.

5 Summary and Future Work

Computer systems and networks have become so essential to the functioning of modern society that they have become a primary target for adversaries, for ideological or nationalistic purposes as an element of modern day warfare, as well as for individual personal or financial gain. Trusted networks are penetrated and exploited to commit espionage and intelligence gathering, perpetrate a denial of service, corrupt data or disseminate misinformation, achieve kinetic or cyber-physical effects, and potentially hijack control of valuable assets. In each vulnerable sector of society, it is essential to prioritize mission essential functions by conducting dependency and risk analysis of network assets, engineering robust and resilient architectures, and identifying the optimal network locations to monitor and swiftly detect compromise of key computer assets.

Many industry and open source tools exist to collect, aggregate, summarize, and organize data collected from hosts and at strategic network locations. Antivirus software and vulnerability scanning tools can systematically search hosts for signatures of malicious code or security flaws in benign code. Intrusion detection and prevention systems can generate alerts that indicate unusual or suspicious activity on computers in the trusted network. The difficulty with these tools is that they generate far too many alerts and indicators, many of which do not truly have security implications. Although an SIEM can be used to assemble, organize and query the data, and help analysts to cope with the large data volumes being generated, existing methods to prioritize the alerts and indicators suffer from various problems and could be improved.

We submit that detecting and comprehending actual threats that exist on the network requires a more dynamic approach. We suggest that correlations between various signatures on a host and indicators of potential exposure to an adversarial entity that may exist in the traffic passing to and from the host might be used to escalate the priority of a known vulnerability. The difficulty is in the sheer volume of signatures and traffic to be processed; it is not a tractable problem for a human being to perform a correlation analysis on each and every set of indicators. We propose certain methods for identifying events of interest, summarizing them, and storing them properly in a database so that cyber analysts can query them easily and instantaneously. Such methods could be combined with SIEM data architectures to provide a more seamless integration with existing methodologies.

The challenge of detecting, assessing, and mitigating vulnerabilities and intrusions necessitates collecting, correlating, and analyzing cyber vulnerability and intrusion data in real-time because

cybersecurity situations evolve rapidly and get complicated with incomplete information and uncertainties. However, current cybersecurity tools and methods have limited capability extensibility and scalability to deal with such complicated situations and big data in real-time. In this chapter, we first presented the basics of vulnerability assessment, data sources and tools, and main components of big data analytics. We then provided a use case on identification and attribution of vulnerability exploitations. Temporal causality analysis of cyber events is described how to determine the quality data needed for the analysis of vulnerabilities and exploitation by determining the temporal interactions and causality of various types of cyber events, including attacker activities, vulnerability detection and protection, and intrusion alerts. This analysis may also assist detecting zero-day vulnerabilities and exploitations whenever the known vulnerabilities and exploits do not provide sufficient reasoning for explaining the suspicious interactions and uncertainties among the observed interactions of attacker activity, vulnerability, and intrusion alerts. For the future research, we suggest that this detection process of zero-day vulnerability and attack be enhanced further by incorporating outlier detection capability into cyber data analytics and causality analysis. To have a better management of cyber events, it would be desirable to add interventions [2] on the values of causality parameters so that the values are not just observed but are also manipulated. In order for cyber analysts to benefit from these scalable data analytics and causality analysis, they should have the capability of forming accurate queries and receiving fast responses by the analytics-driven processing environment of cybersecurity.

References

1. E. Cole. “Detect, Contain and Control Cyberthreats”. A SANS Whitepaper, SANS Institute, June 2015.

2. D. Koller and N. Friedman. “Probabilistic Graphical Models”. MIT Press, 2009.

3. A. Vijayakumar and G.A. Muthuchelvi. “Discovering vulnerability to build a secured system using attack injection.” *2011 3rd International Conference on Electronics Computer Technology (ICECT)*, Vol. 6. IEEE, 2011.

4. H. Zhang, D. Yao, N. Ramakrishnan, and Z. Zhang. “Causality reasoning about network events for detecting stealthy malware activities.’’ *Computers and Security*, 58, 2016, pp. 180-198.

5. A. Kim, M.H. Kang, J.Z. Luo, and A. Velazquez. “A Framework for Event Prioritization in Cyber Network Defense.’’ Technical Report, Naval Research Laboratory, July 15, 2014.

6. S.M. Sawyer, T.H. Yu, M.L. Hubbell, and B.D. O’Gwynn. “LLCySA: Making Sense of Cyberspace.’’ Lincoln Laboratory Journal, Vol. 20, No. 2, 2014, pp. 67-76.

7. FIRST: Improving Security Together. Common Vulnerability Scoring System (CVSS-SIG). Available from: http://www.first.org/cvss.

8. National Vulnerability Database. NVD Common Vulnerability Scoring System Support v2. Available from: http://nvd.nist.gov/cvss.cfm.

9. P. Mell, K. Scarfone, and S. Romanosky. “CVSS – A Complete Guide to the Common Vulnerability Scoring System Version 2.0.” June 2007.

10. K. Scarfone and P. Mell. “An Analysis of CVSS Version 2 Vulnerability Scoring’’. *Proceedings of IEEE 3rd International Symposium on Empirical Software Engineering and Measurement*, 2009.

11. H. Cam. "Risk Assessment by Dynamic Representation of Vulnerability, Exploitation, and Impact.'' *Proceedings of Cyber Sensing 2015, SPIE Defense, Security, and Sensing*. April 20-24, 2015, Baltimore, MD.

12. AlienVault Unified Security Management. Available from: https://www.alienvault.com/products/.

13. AlienVault Unified Security Management. Available from: https://www.alienvault.com/doc-repo/USM-for-Government/all/Correlation-Reference-Guide.pdf.

14. AlienVault Unified Security Management. Available from: https://www.alienvault.com/open-threat-exchange.

15. Enterprise Security Management (ESM). Available from: http://www8.hp.com/us/en/software-solutions/arcsight-esm-enterprise-security-management/.

16. HP. Protect 2104. Available from: http://h71056.www7.hp.com/gfs-shared/downloads-220.pdf.

17. Available from: http://www.hp.com/hpinfo/newsroom/press_kits/2011/risk2011/HP_ArcSight_Express_Product_Brief.pdf.

18. Hewlett Packard Enterprise. Available from: http://www8.hp.com/h20195/V2/GetPDF.aspx/4AA4-3483ENW.pdf.

19. HP. Available from: http://www.hp.com/hpinfo/newsroom/press_kits/2015/RSA2015/ThreatCentralDataSheet.pdf.

20. US-CERT. “Information Sharing Specifications for Cybersecurity.” Available from: https://www.us-cert.gov/Information-Sharing-Specifications-Cybersecurity.

21. Github. Available from: https://stixproject.github.io/supporters/.

22. HP Inc. “HP to Acquire ArcSight”. Available from: http://www8.hp.com/us/en/hp-news/press-release.html?id=600187#.V1zA5Kpf1R0.

23. Kahn Consulting Inc. Available from: http://www.kahnconsultinginc.com/images/pdfs/KCI_ArcSight_ESM_Evaluation.pdf.

24. Splunk. Available from: http://www.splunk.com/view/SP-CAAAGBY.

25. Splunk. http://www.splunk.com/en_us/products/premium-solutions/splunk-enterprise-security.html

26. Splunk. Available from: http://docs.splunk.com/Documentation/Splunk/latest/Search/Aboutthesearchlanguage.

27. Splunk. Available from: https://conf.splunk.com/session/2015/conf2015_ANekkanti_SPal_ATameem_Splunk_SplunkClassics_Harnessing63PerformanceAnd_a.pdf.

28. HP. Available from: http://www8.hp.com/h20195/v2/GetPDF.aspx/4AA5-2823ENW.pdf

29. Baker, George H. “A Vulnerability Assessment Methodology for Critical Infrastructure Sites”. *DHS Symposium: R&D Partnerships in Homeland Security*, 2005.

30. GAO: Government Accountability Office. Defense Department Cyber Efforts: DOD Faces Challenges in Its Cyber Activities, 62 (GAO-11-75), July 2011. Available from: http://itlaw.wikia.com/wiki/GAO.

31. R. Schwarz and F. Mattern, “Detecting Causal Relationships in Distributed Computations: In Search of the Holy Grail.” *Distributed Computing*, March 1994, Vol. 7, No. 3, pp. 149-174.

32. AlienVault. Available from: https://www.alienvault.com/products/ossim.

33. Enterprise Security Management (ESM). http://www8.hp.com/us/en/software-solutions/arcsight-esm-enterprise-security-management/.

34. Splunk. Available from: https://www.splunk.com/en_us/.