Available online at www.sciencedirect.com

ScienceDirect

journal homepage: www.elsevier.com/locate/cose

Computers & Security

A novel kill-chain framework for remote security log analysis with SIEM software

Blake D. Bryant, Hossein Saiedian *

Electrical Engineering and Computer Science, University of Kansas, Lawrence, KS 66045, USA

ARTICLE INFO

Article history:
Received 25 February 2016
Received in revised form 14 February 2017
Accepted 4 March 2017
Available online 8 March 2017

Keywords:
Kill-chains
Cyber ontology
Cyber forensics
Computer security
Remote logging
SIEM
Intrusion detection alerts
Operating system audit logs

ABSTRACT

Network security investigations pose many challenges to security analysts attempting to identify the root cause of security alarms or incidents. Analysts are often presented with cases where either incomplete information is present, or an overwhelming amount of information is presented in a disorganized manner. Either scenario greatly impacts the ability for incident responders to properly identify and react to security incidents when they occur. The framework presented in this paper draws upon previous research pertaining to cyber threat modeling with kill-chains, as well as the practical application of threat modeling to forensic. Modifications were made to conventional kill-chain models to facilitate logical data aggregation within a relational database collecting data across disparate remote sensors resulting in more detailed alarms to security analysts. The framework developed in this paper proved effective in identifying the relationship of security alarms along a continuum of expected behaviors conducive to executing security investigations in a methodical manner. This framework effectively addressed incomplete or inadequate alarm information through aggregation, and provided a methodology for organizing related data and conducting standard investigations. Both improvements proved instrumental in the effective identification of security threats in a more expeditious manner.

1. Introduction

Information security is one of the primary concerns for executives leading organizations across the globe, with increasing visibility among key decision makers each passing year (PWC, 2015). Media coverage of high profile security breaches has exacerbated concerns that current strategies to detect and prevent intrusions are inadequate. Some researchers argue that this inadequacy is derived from reliance on statistical models or attack signatures that attackers have learned to avoid (Beaver et al., 2013; Chen and Malin, 2011; Ioannou et al., 2013; Shiva et al., 2010). Other researchers argue that point solutions may be capable of detecting certain aspects of attacks actions, but cannot provide all data required to verify malicious activity due to their inability to observe all events that occur across the network affecting multiple computing systems (Claycomb and Shin, 2010). Due to these limitations, it has become increasingly important to analyze data from multiple auditing systems deployed in separate locations within a network topology in order to detect sophisticated attacks (Best et al., 2014; Claycomb and Shin, 2010; Flagg et al., 2007; Ioannou et al., 2013; Ross et al., 2011; Rush et al., 2015; Shalyapin and Zhukov, 2015). Unfortunately, aggregation of data from multiple sources presents several challenges such as managing large volumes of data from disparate sensors and making sense of disorganized, incompatible, or seemingly chaotic data (Beaver et al., 2013; Best et al., 2014; Chen and Malin, 2011; Claycomb and Shin, 2010; Shiva

* *Corresponding author.*
E-mail address: saiedian@eecs.ku.edu (H. Saiedian).
http://dx.doi.org/10.1016/j.cose.2017.03.003

et al., 2010). These challenges greatly impact information security and incident response teams' ability to identify security breaches or implement corrective action to detect, prevent or mitigate the damage incurred during attacks.

This paper focuses on overcoming these challenges by implementing a mechanism for fusing multi-sensor data into machine generated alerts that accurately depict attack actions across multiple disparate sensors, ultimately resulting in improved detection rates and streamlined analyst workflows. Analysis of historical security breach data and successful security analyst investigative techniques has yielded insights toward effective threat identification via behavioral modeling (Hutchins et al., 2010; Ioannou et al., 2013; Kul and Upadhyaya, 2015; McWorther, 2013; Rush et al., 2015; Shalyapin and Zhukov, 2015). Research in threat behavioral modeling resulted in the adoption of the term "kill-chain," borrowed from United States military doctrine for targeting individuals of interest through behavioral analysis, as a means to describe the sequence of events an attacker must perform in order to achieve success during an attack (Hutchins et al., 2010).

This paper was inspired by the Lockheed Martin Kill-Chain, which is currently used by the National Institute of Standards and Technology (NIST) as a component of the Cyber Security Framework, and is often cited as the original cyber kill-chain model (Hutchins et al., 2010). However, attempts to implement the Lockheed Martin Kill-Chain within production environments as a mechanism to aggregate or correlate data from disparate sensor systems proved to be a difficult, if not impossible, task. This led to the hypothesis that additional phases would be required to properly address the challenges imposed by variations in metadata generated within the generic phases of the kill-chain. This paper presents a novel kill-chain model with specific phases designed to facilitate meta-data aggregation in order to overcome the challenges of incomplete or disconnected sensor data presented by other security researchers.

Application of this framework within a security information and event management (SIEM) system proved that the additional kill-chain phases were easily adaptable to the existing correlation engine resulting in improved alerts that outperformed individual sensor false positive and false negative rates. Additionally, alarms generated via this correlation framework resulted in a drastic reduction in alarm noise and volume while simultaneously providing fused data conducive to more efficient investigation workflows and rapid threat identification by security analysts.

2. Kill-chain models

The Lockheed Martin Kill-Chain consists of seven phases designed to represent attacker objectives that should be accomplished in order to successfully compromise a targeted network and perform malicious actions, such as data theft, denial of service, or system destruction. Security researchers have been able to identify empirical evidence for most of the phases within the Lockheed Martin Kill-Chain and attribute said evidence to indicators of attempts to achieve the attacker objectives defined by their respective phases.

However, these phases often extend beyond the scope of a single organization's network and may require data unavailable to internal security teams to identify threats. The first two phases of the kill-chain, as defined by Hutchins et. al., pertain to reconnaissance of potential victims' security vulnerabilities and the development of tools to exploit said vulnerabilities before attempting to attack the victim's network (Hutchins et al., 2010). These vulnerabilities may exist in the form of technical or non-technical components of a victim's network, such as public information pertaining to the identities of executive leadership, identifying non-technical vulnerabilities associated with target identification; or job postings for individuals trained on specific information systems, identifying systems that may be targeted with existing exploits or will require weaponization.

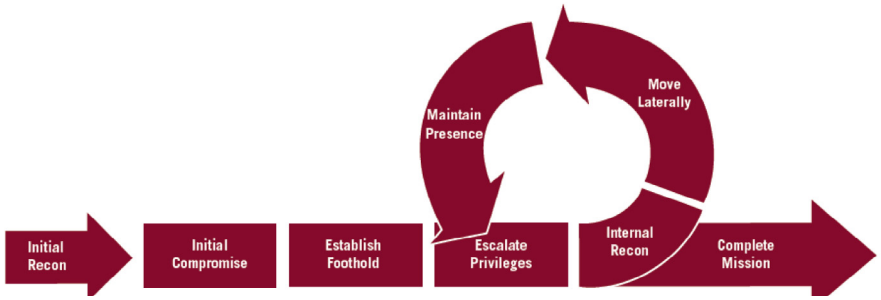
Weaponization may entail the development of custom applications, or may consist of combining existing technologies with privileged information, such as the combination of a publicly available exploit for Microsoft Office products delivered to a member of the victim's executive staff via a phishing email. Additionally, the "reconnaissance" phase of the Lockheed Martin model is often incorrectly attributed to reconnaissance actions performed within the victim's network. Undoubtedly, reconnaissance actions will be observed when performing analysis of network security incidents; however, these activities are not easily delineated from the "delivery" phase described in the Lockheed Martin model.

Likewise, exploitation is similarly defined in ambiguous terms as exploits may occur in the form of network delivered payloads or client side vulnerabilities. These ambiguities require data scientists and security analysts to make judgment calls when deciding how to categorize data associated with these activities, which often results in inconsistent data classification and meta data parsing. Fig. 1(a) below depicts the Lockheed Martin Kill-Chain.

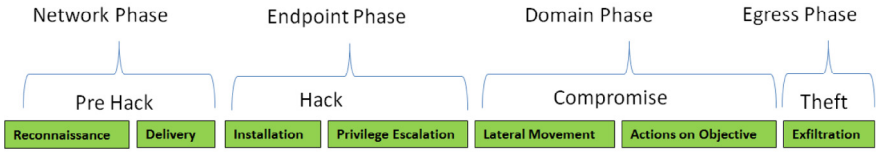
Another prominent kill-chain model is the Mandiant Attack Lifecycle published in 2012 (McWorther, 2013). The Mandiant model differs from the Lockheed Martin model by focusing on internal network activities, lending to more direct application to security analysis use cases than the generic Lockheed Martin model. Additionally, the Mandiant model accounts for recursive internal reconnaissance and lateral movement activities often exhibited by attackers following the initial breach. Though the Mandiant model arguably presents a better model of the actions performed by attackers during a security breach, it still lends itself to interpretation of what indicators are likely to be attributed to each action group. This again results in inconsistent data analysis ultimately leading to less efficient workflows by security personnel. Fig. 1(b) below depicts the Mandiant Attack Lifecycle.

A new kill-chain model was devised in order to capitalize on the phased analysis approach lauded by both of the previous models, but within the context of organizing data from each phase into a structured database to support data queries and correlation routines. The new kill-chain was constructed consisting of a seven phase model, similar to the Lockheed Martin model; however, some phases from the Lockheed Martin model were omitted or shifted within the sequence of events and two new phases were introduced from the Mandiant model. The Lockheed Martin "weaponization" phase was omitted as

(a) Lockheed Martin Kill-Chain[13]



(b) Mandiant Attack Lifecycle[14]



(c) Bryant Kill-Chain

Fig. 1 – Three kill-chain models: (a) Lockheed Martin Kill-Chain, (b) Mandiant Attack Lifecycle, and (c) Bryant Kill-Chain.

it often occurs outside the victim's network and is not likely to be observed within sensor data. The delivery phase was shifted to occur immediately following reconnaissance and the installation phase was shifted to occur after the delivery phase. The exploitation phase was replaced by the delivery, installation and privilege escalation phases respectively, as components of exploits, e.g. exploit delivery detected by an IDS, exploit installation detected by a malware detection engine, or privilege process execution resulting from malicious code injection, are often observed in these phases with no other unique identifiers warranting a dedicated exploitation phase. The command and control phase was omitted as the phases of delivery, installation, lateral movement or exfiltration exhibited evidence of command and control based on the tools/techniques used. The privilege escalation phase from the Mandiant model was added in order to segregate privileged account data from routine user data and facilitate identification of improper or anomalous use of elevated credentials. The lateral movement phase from the Mandiant model was introduced to differentiate between reconnaissance activity originating from an external network and reconnaissance activity from an internal network. Finally, the exfiltration phase was added in order to provide specific emphasis on anomalous data transfers originating from the internal network to an external network.

The new kill-chain phases were also designed to align with natural identifiers leveraged by analysts when performing investigations within each phase. Reconnaissance activity was often investigated based on the originating IP address, with data generally reflecting a one to many relationships between the source IP address and multiple destination IP addresses being reconnoitered. Data within the delivery phase often exhibited a single source IP address targeting a single destination IP address and enumerating through a large number of potential exploits. Installation activity was often investigated based on the computer name of an infected machine, as IP address data were often omitted from systems used to detect malware or software modifications. Data within the privilege escalation phase were investigated based on user credentials and often exhibited a newly created administrator attempting to perform multiple command line actions. Lateral movement was also investigated based on the user credentials used and often exhibited a single user attempting to access multiple different machines. Actions on the objective were most commonly investigated by the computer name of the system suspected of compromise in order to determine the nature of changes to the system. Finally, the egress phase was investigated by analysis of unusual foreign destination IP addresses. The ability to apply a natural identifier to each phase during investigations was assessed to be an advantage over existing kill-chain models which occasionally resulted in inconsistent investigation methodologies.

Forensic data were analyzed from a pool of historical data breaches and sanctioned penetration tests conducted by third parties in order to identify specific indicators of activity within each phase of the new model. Analysis indicated distinct traits identifiable in data extracted from four distinct macro phases: network, endpoint, domain, and egress. Data extracted from each of these phases could be further deconstructed into sub-phases based on attacker actions or anomalous behaviors observed in data. Fig. 1(c) below depicts the new model which

will be referred to as the Bryant Kill-Chain Model throughout the remainder of the paper.

3. Application of Bryant Kill-Chain model in network forensics

The Bryant Kill-Chain model was applied as a forensics approach to additional third party penetration tests or postmortem analysis of suspected security breaches in order to validate the efficacy of implementing the model within an automated correlation system. In most cases, data analyzed during this evaluation identified gaps or inconsistencies in auditing configurations by the affected parties, resulting in partial detections of attacker activities. Partial or omitted detections indicated that strict enforcement of phase sequencing during automated correlation would likely be infeasible in most network environments. However, plotting detected activities along the spectrum of the kill-chain proved to be an effective tool for rapidly identifying gaps as well as evaluating the maturity of disparate security organizations or architectures.

Organized and consistent analysis of data via this framework uncovered several natural patterns in forensic data conducive to future breach discovery work as specific indicators of compromise were consistently presented within each phase despite differences in security sensor manufacture. Additionally, analyst suspicions of activity within a discrete phase could be evaluated with prefabricated stored procedures or queries for all expected data within a related phase, revealing attacker actions and confirming the lack of data due to improper audit policies, inadequate sensor configuration, or suboptimal sensor placement/architecture. Consistent patterns of data recurrence enabled analysts to raise alarm when expected indicators were not observed prompting audit policy modifications or sensor configuration changes that often produced the expected data once implemented. The initial seven phases of the Bryant Kill-Chain were deconstructed into 13 subtasks in order to reflect slight variations between data and attacker behaviors. Fig. 2 below depicts the process used by analysts to plot metadata assessed to be indicators of compromise along the Bryant Kill-Chain.

During postmortem analysis of security breaches, analysts who leveraged the Bryant Kill-Chain as an investigative framework consistently provided more thorough data and analysis to investigators when compared to their peers implementing an ad-hoc analytical methodology. Likewise, stakeholders often responded positively to using the model as a visual representation depicting where breaches were discovered across an event spectrum. Communication of potential mitigation strategies or security solutions was also simplified by referring to the "sensor" section of the model. The improvements in analyst workflow products, process, and stakeholder reception were deemed to be adequate to validate the model's efficacy for inclusion within SIEM software.

4. Applying the kill-chain to SIEM software

The LogRhythm® SIEM platform was selected as the preferred system to evaluate inclusion of a kill-chain model based on the author's prior experience with the system and access to historical data conducive to evaluating multiple production environments. The IBM Qradar®, McAfee Nitro® and Splunk® platforms also exhibited potential to be modified to incorporate this model, but were not evaluated within this paper. Fig. 3 below depicts data flow as sensor information is transformed into alarms within the LogRhythm® SIEM.

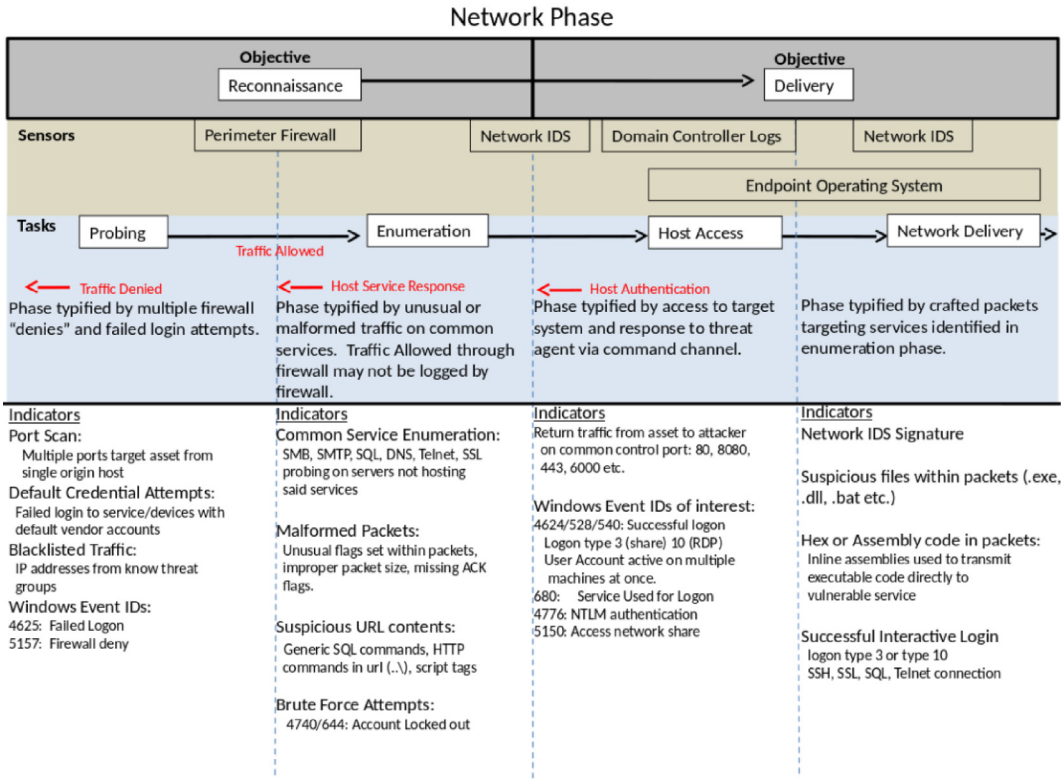
The LogRhythm® dataflow model implements suspicion escalation and data triage functions by parsing sensor information into a threat ontology and applying descriptive "classification" labels to observed events. The classification label is potentially applied in two different stages of the data flow model; either by the message processing engine, during the initial parsing and normalization phase, or by the advanced intelligence engine, during correlation and subsequent reclassification. Classification labels serve a unique function as they introduce new metadata into an event record that was not present within the raw log information. This provides a mechanism for combining previously dissimilar data from disparate sensors into corroborating data sets. The classification field was determined to be the ideal candidate for implementing the new kill-chain phased model within the SIEM as this would facilitate rapid identification of related events and enable future event correlation during alarming.

Deconstructing the Bryant kill-chain and analyzing metadata within each sub-phase yielded insight to potential data parings for correlation. Each sub-phase was evaluated for suitability as a table within a relational database and metadata within each phase was evaluated for suitability as primary or foreign keys to be used to join adjacent sub-phase data as the LogRhythm SIEM utilized SQL queries to perform correlation functions. Fig. 4 below illustrates the primary and foreign key relationship, with the primary key identified by a red box and the foreign key identified by a light blue line to the adjacent phases.

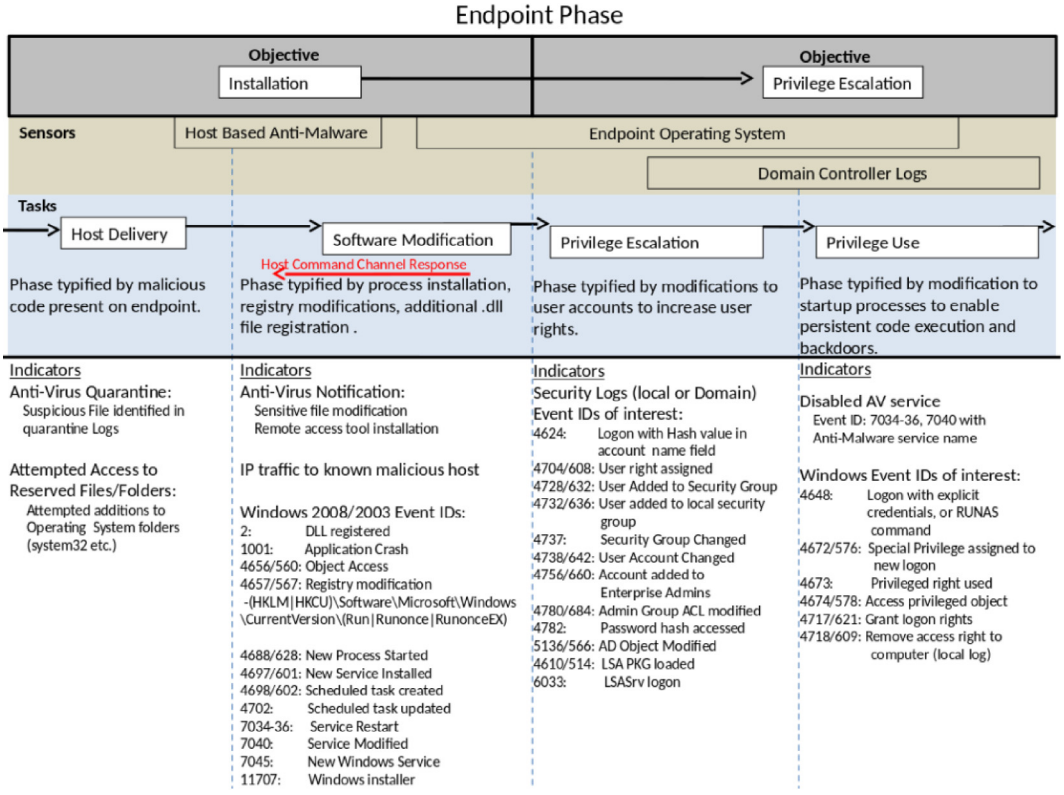
Of note, some logging systems did not provide enough data for correlation with adjacent phases without fusing data with another source within the same phase. Correlating network delivery events with installation events is a prime example of this phenomenon. Network intrusion detection systems often omit the host name of machines, while host based malware solutions often contain the hostname but omit the local IP address. This issue was resolved by fusing both data sets via DHCP, DNS, or domain authentication data available on domain controllers or servers hosting these common services.

Sensor logs, SIEM events or security alarms were aggregated within each phase with SQL queries joining metadata via classification field and the phase specific aggregate field depicted in Fig. 4. For example, all events classified as being associated with the reconnaissance objective group (reconnaissance, probing or enumeration) with the same source IP address were aggregated within a single event. In order to provide the maximum forensic value to analysts, the aggregate event retained all unique meta-data fields observed within aggregated records.

In most cases, the resultant aggregate event exhibited drastically improved forensic value to security analysts, especially for attack activities that typically generate a large volume of logs on sensor devices. A prime example of this phenomenon is reconnaissance activity associated with network

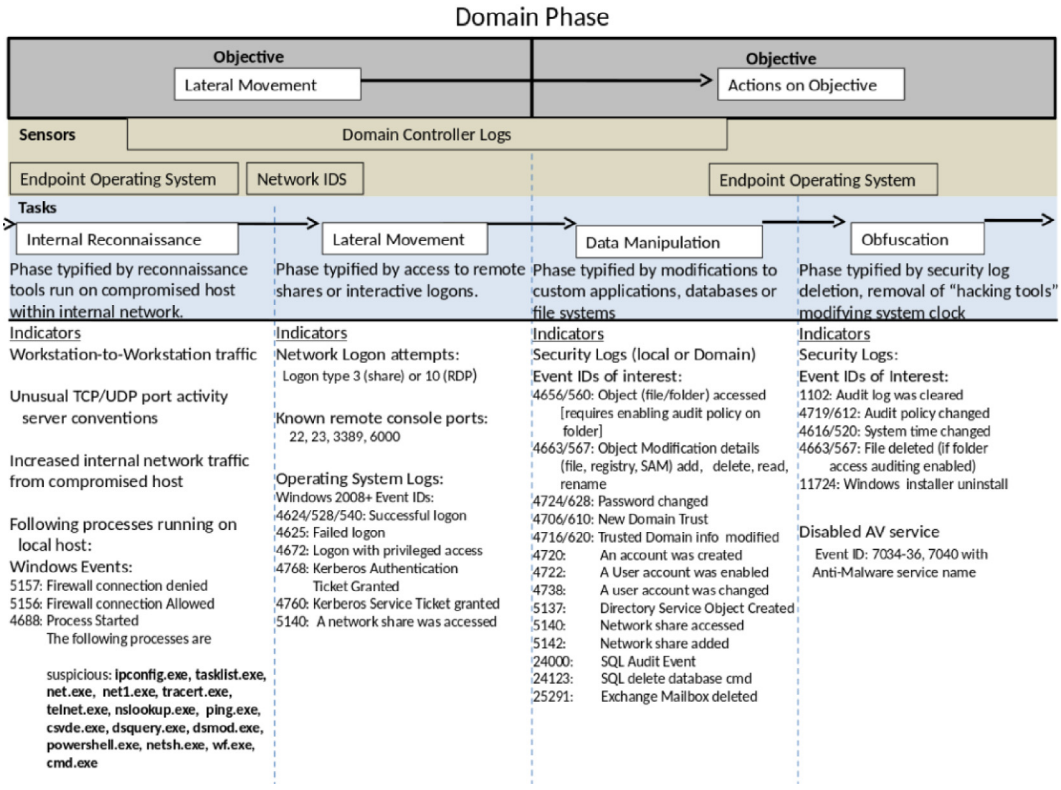


(a) Deconstructed Network Phase

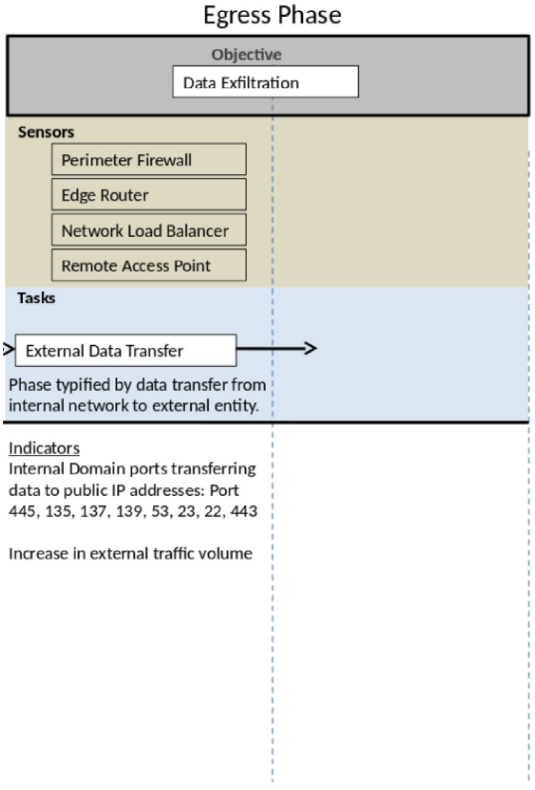


(b) Deconstructed Endpoint Phase

Fig. 2 – Bryant Kill-Chain as an investigation framework.

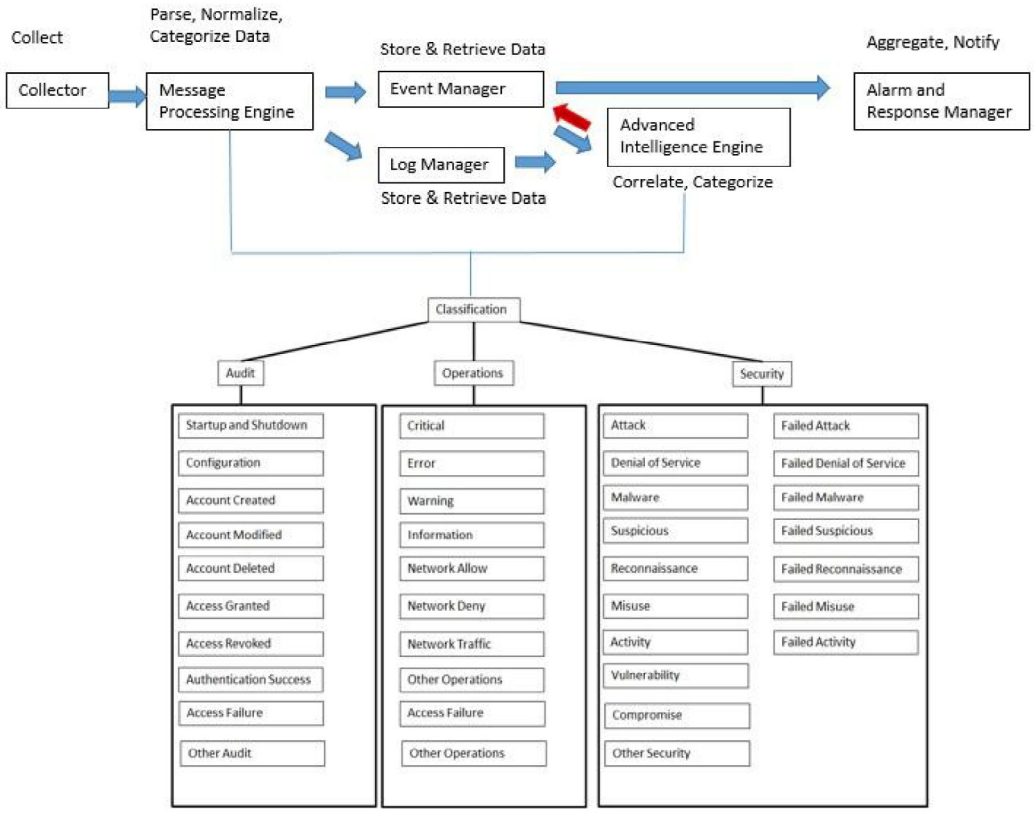


(c) Deconstructed Domain Phase

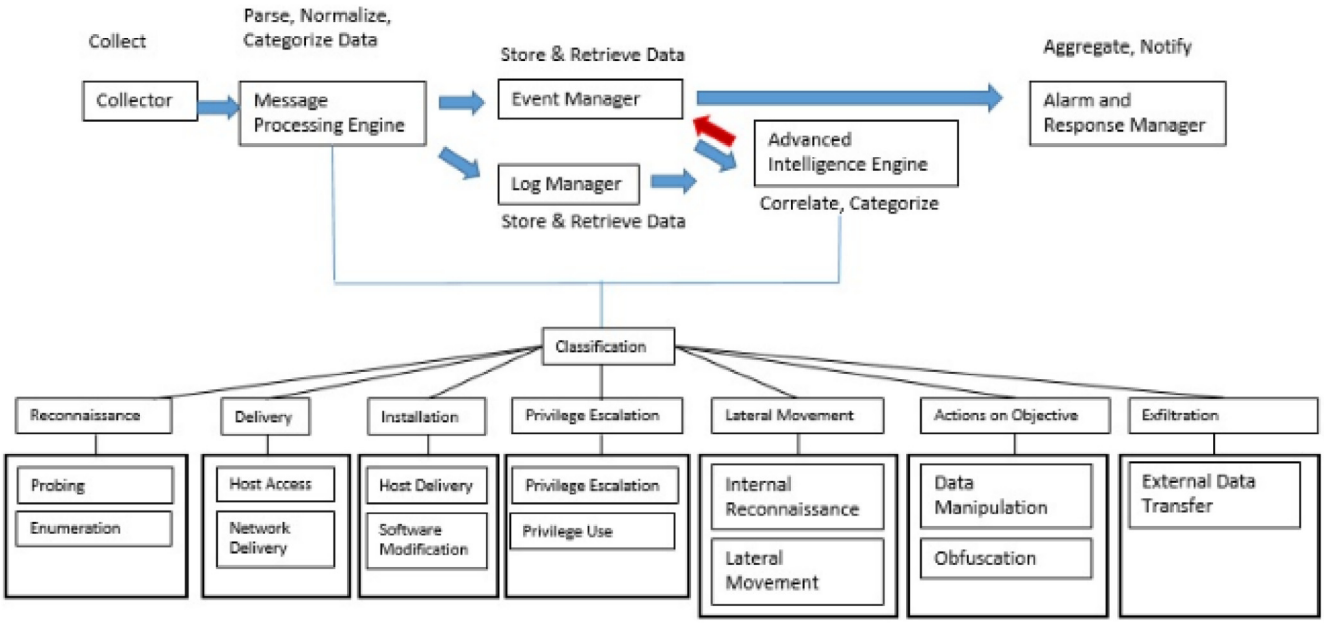


(d) Deconstructed Egress Phase

Fig. 2 – (*continued*)



(a) Data Flow Base Ontology



(b) Data Flow Modified Ontology

Fig. 3 – LogRhythm® data flow: (a) data flow base ontology and (b) data flow modified ontology.

scanning tools. During the evaluation, a single aggregate alarm was generated from 100 individual events observed during a vulnerability scan. The resultant alarm fused information from multiple logging systems and accurately identified all target machines affected by the attacker, as well as all unique exploits or signatures observed by logging devices. Fig. 5 below depicts a screen shot of alarms generated during an OpenVAS vulnerability scan using the baseline SIEM configuration and

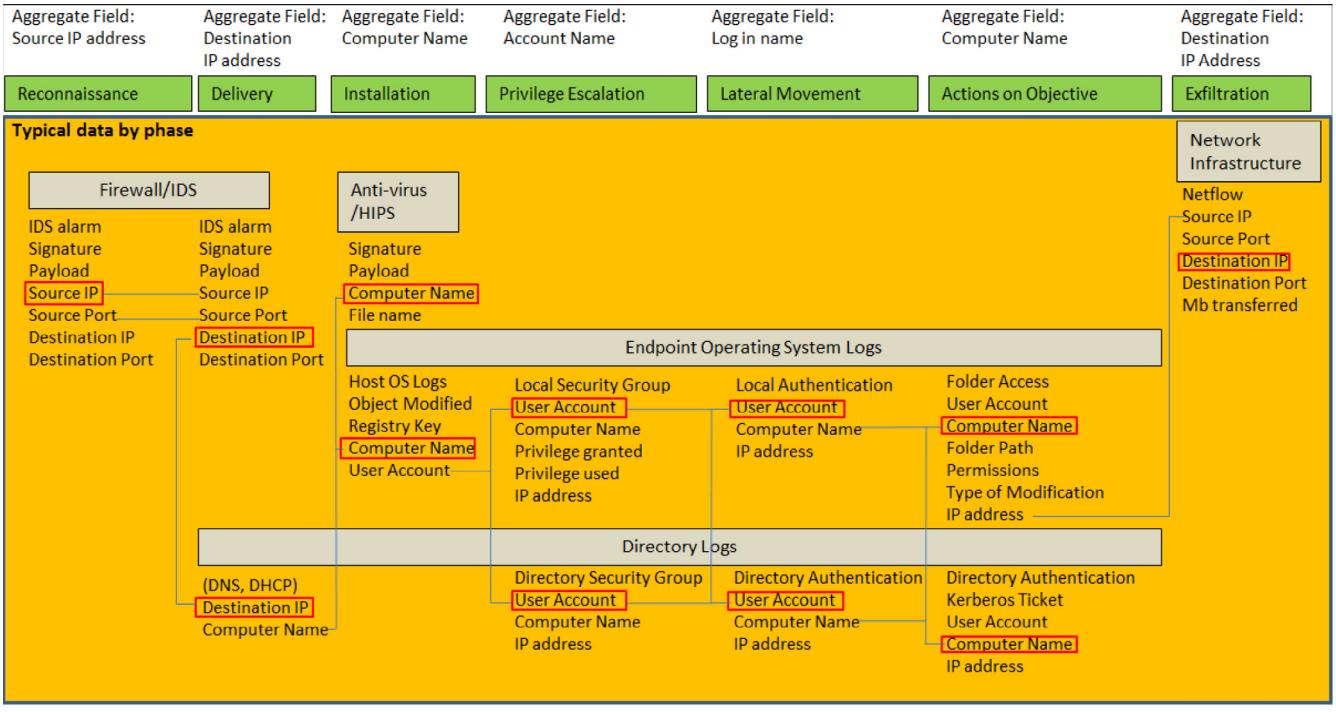


Fig. 4 – Data relationships for correlation.

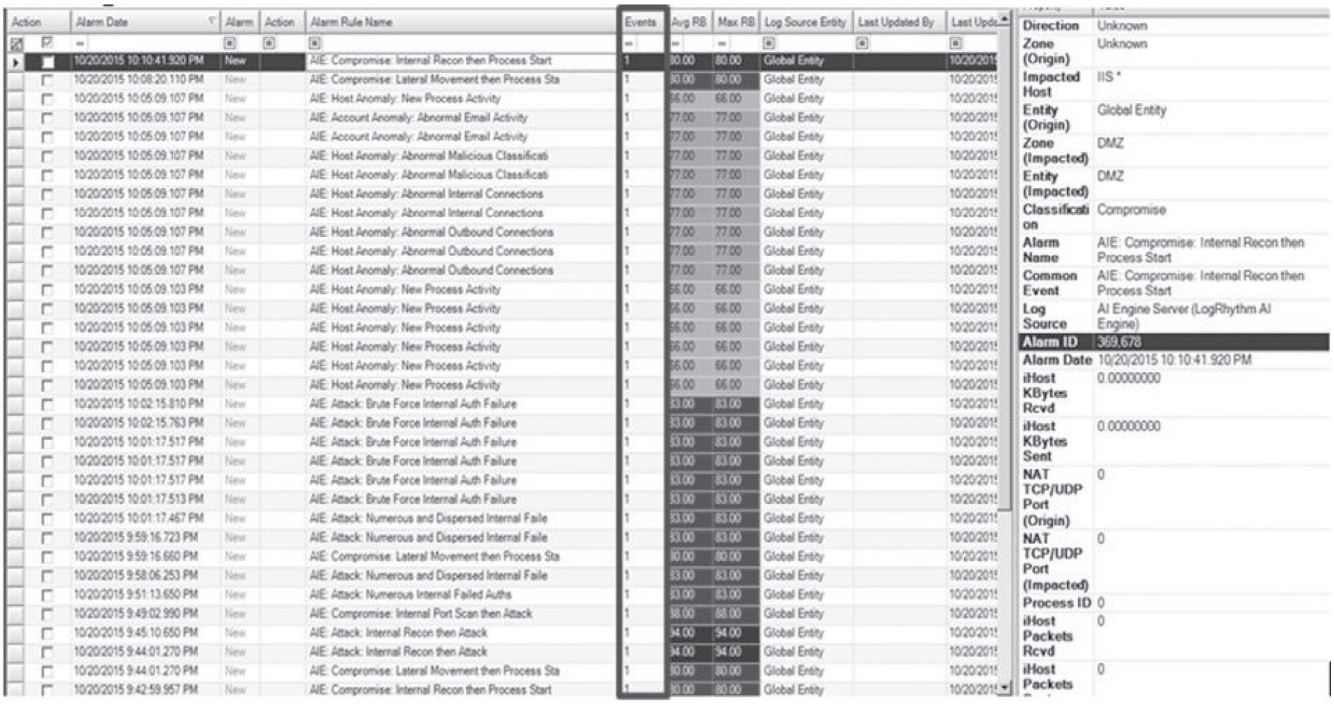
the modified SIEM configuration. The baseline SIEM generated 41 alarms from 41 normalized events, while the modified SIEM generated 5 alarms from 401 normalized events. The modified alarms contained distinct metadata from up to 100 events in each alarm providing insight pertaining to the number of target hosts probed, signatures detected, ports used, and kill-chain phases traversed.

5. Evaluation of SIEM detection rates with new framework

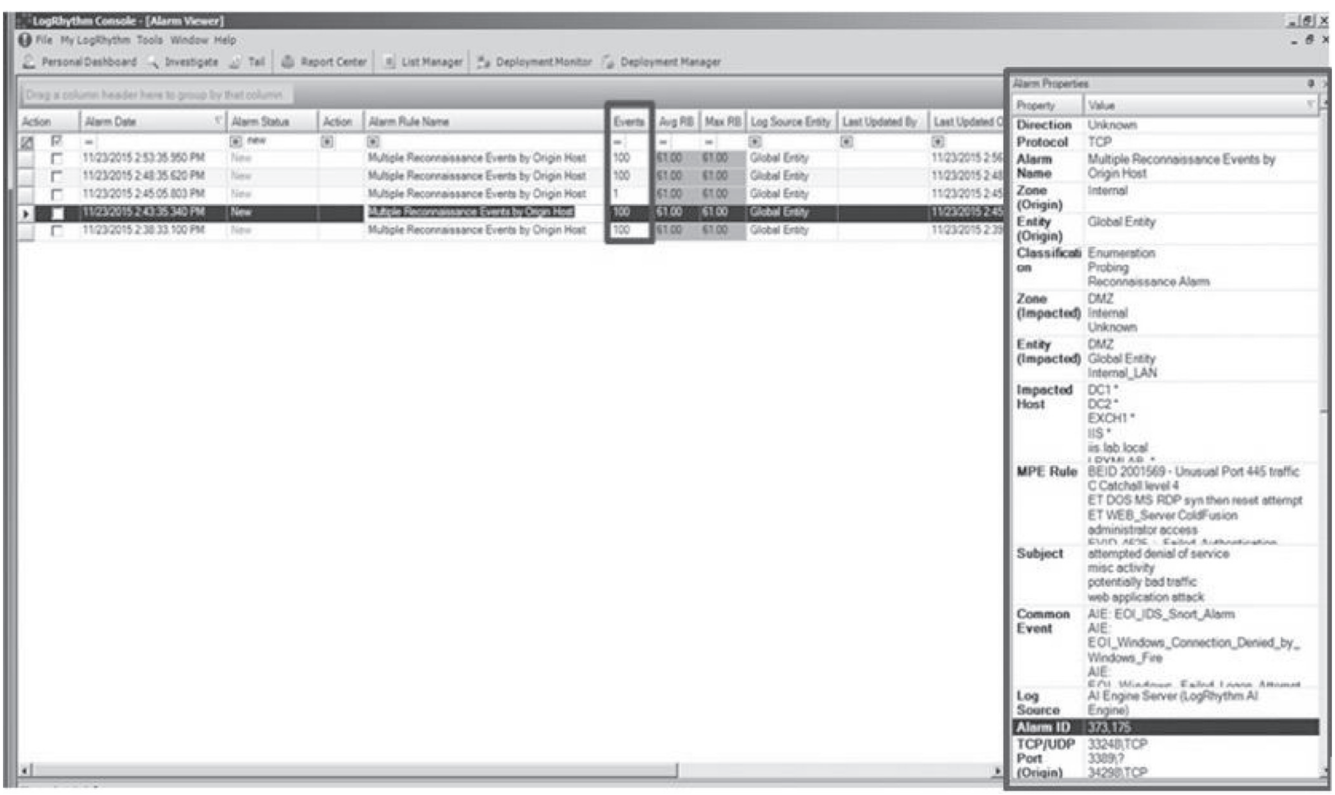
A virtual network was constructed in order to compare the quality and quantity of alarms generated by the baseline SIEM configuration and a modified SIEM configuration implementing the Bryant Kill-Chain. Fig. 6 below depicts the virtual network topology used during testing. One Logrhythm SIEM system was installed within a virtual environment and configured with vendor recommended default correlation rules. The virtual machine was then cloned and modified to incorporate the new framework and new correlation rules leveraging the indicators of compromise discovered via application of the Bryant Kill-Chain to postmortem incident analysis. The virtual environment network consisted of two Microsoft Windows domain controllers, a Microsoft Exchange mail server, a Microsoft SharePoint server, a Windows 7 workstation, a Linux based Snort intrusion detection system, a Linux based pFsense firewall, McAfee© host based intrusion prevention system and anti-malware software installed on all endpoints, and an attacker machine configured with the Kali Linux penetration testing suite of tools installed. Traffic was segregated into multiple subnets by the pFsense firewall with the Snort IDS platform resident on the same operating system as the firewall. Email and web services were hosted from within a segregated DMZ, domain controllers and workstations resided within a simulated protected LAN, and the attacker machine was connected to a simulated external LAN interface on the firewall.

A custom attack scenario was created to stimulate all seven of the attack phases, across multiple machines, generating data from multiple detection sensors, and ultimately resulting in successful data theft. A custom scenario was required as certain attacks do not require complete attack cycle completion, such as commodity malware that may execute delivery, installation and exfiltration phases only; or attacks may be eliminated by security systems, such as exploits interdicted by IPSs or malware removed by host based security suites. The following paragraphs provide a brief summary of the test scenario which encompassed the 26 test cases depicted in Fig. 7 and Table 1 in the conclusions section.

The custom scenario involved an attacker located on an external 172.16.x.x network attempting to access an internal 10.x.x.x network simulating a small corporate environment. The attacker performed initial probing activities to identify victim systems, determined that all systems probed were not susceptible to generic exploits due to recent patches, or were blocked by firewall or IPS policies. The attacker decided to send a phishing email through the corporate mail server in an attempt to trick a user into installing a legitimate program with a known vulnerability and a custom backdoor disguised as a patch to the vulnerable program. The fake patch was actually a custom python application that established a remote shell to the attacker machine. The fake patch was not detected by host based malware signatures because it was a simple socket program that executed command redirection and had no known signature; however, it did generate command execution evidence within operating system logs. The attacker leveraged the



(a) Alarms Associated with OpenVAS Scan Using Baseline SIEM Ontology



(b) Alarms Associated with OpenVAS Scan Using Modified SIEM Ontology

Fig. 5 – LogRhythm alarm examples.

Table 1 – SIEM alarm evaluation results.

Test case	Case name	Baseline alarms	Baseline events	Modified alarms	Modified events	Raw logs
1	Nmap Port Scanning	0	0	1	100	87
2	SMB Scan	0	0	0	0	76
3	Open Vas Vulnerability Scan	41	41	5	401	4158
4	Phishing Email	1	1	1	1	92
5	Suspicious Download	0	0	1	1	25
6	Unauthorized Software Installation	0	0	2	18	105
7	Python Reverse Shell	0	0	2	3	344
8	Privilege Escalation New Local Admin	3	3	1	6	997
9	Remote Desktop From Kali to Windows	0	0	2	3	174
10	Disable anti-virus	0	0	1	3	86
11	Launch Meterpreter Reverse Shell	18	18	1	1	106
12	Hash Extraction	0	0	1	3	55
13	Network Share Creation	0	0	3	6	33
14	Internal Reconnaissance Tools	0	0	1	1	54
15	Pass the Hash to Webserver	0	0	3	27	80
16	Copy SQL Database	0	0	2	8	250
17	Privilege Escalation New Local Admin	1	1	2	23	61
18	Remote Desktop Workstation to Webserver	0	0	4	11	353
19	Internal Data Transfer Webserver to Workstation	0	0	1	2	64
20	Pass the Hash to Webserver	0	0	1	1	51
21	Privilege Escalation New Local Admin	1	1	1	8	64
22	Copy Email Database	0	0	1	12	131
23	Remote Desktop Workstation to Email Server	0	0	4	10	204
24	Internal Data Transfer Email Server to Workstation	0	0	1	5	80
25	External Data Transfer Workstation to Kali	18	18	1	1	56
26	Audit Log Purging	0	0	3	11	304

weak shell presented by the python program to conduct privilege escalation by replacing a service installed by the legitimate program contained within the phishing email with a custom executable program containing windows "net user" commands to create a local administrative account within the remote desktop users group on the target machine. The injected executable was executed with system privileges upon a system restart, as prescribed in the phishing email. The attacker then initiated a remote desktop session on the compromised machine and disabled the McAfee host based intrusion detection system and anti-virus software in order to upload hacking tools for hash extraction and a reverse meterpreter shell. The attacker extracted domain administrator hash values from the compromised machine, executed the reverse meterpreter shell, and copied the hash values into the meterpreter console of the attacking machine for use in pass-the-hash techniques later on. The attacker continued to use the open remote desktop session on the compromised machine to create a network share to be used for staging stolen data prior to the final exfiltration from the network, and conducted ping, traceroute and NSLOOKUP commands from the compromised machine to reconnoiter the corporate network. The attacker identified the internal IP address of the company webserver, email server, and domain controllers from the internal reconnaissance and used the meterpreter shell to log onto the webserver using the extracted hash associated with the domain administrator account.

The attacker leveraged the remote shell established via the pass-the-hash program to execute a series of SQL commands on the webserver in order to identify the SQL database used to host the corporate web content and leveraged SQL backup routines to create an offline copy of the database to the local c: drive of the web server. The attacker then created a new local administrative account within the remote desktop users group on the webserver to be leveraged as a backdoor in case the initial foothold was eliminated. The attacker then established a remote desktop session from the initially compromised workstation to the webserver in order to validate the new backdoor account, and copy the offline database to the network share previously created on the compromised workstation. This step was designed to stimulate the lateral movement test case and may be omitted in a legitimate breach scenario.

The attacker then conducted pass-the-hash to access the corporate email server, created a local administrator backdoor account with remote access, and traversed the windows server directory until it located the exchange mailbox *.edb file. The attacker then stopped the mail service with the "net stop" command and copied the database to the server's local hard drive to be staged for an internal transfer to the compromised workstation. The attacker used the initially compromised workstation to establish a remote desktop session to the email server leveraging the backdoor account and copied the email database to the network share on the workstation. The attacker then copied the email database and SQL databases from the compromised workstation to a SAMBA share hosted on the attacker's Kali Linux machine. Finally, the attacker executed the command "wevtutil cl application && wevtutil cl security && wevtutil cl system" from existing pass-the-hash consoles on each affected system to purge the operating system audit logs.

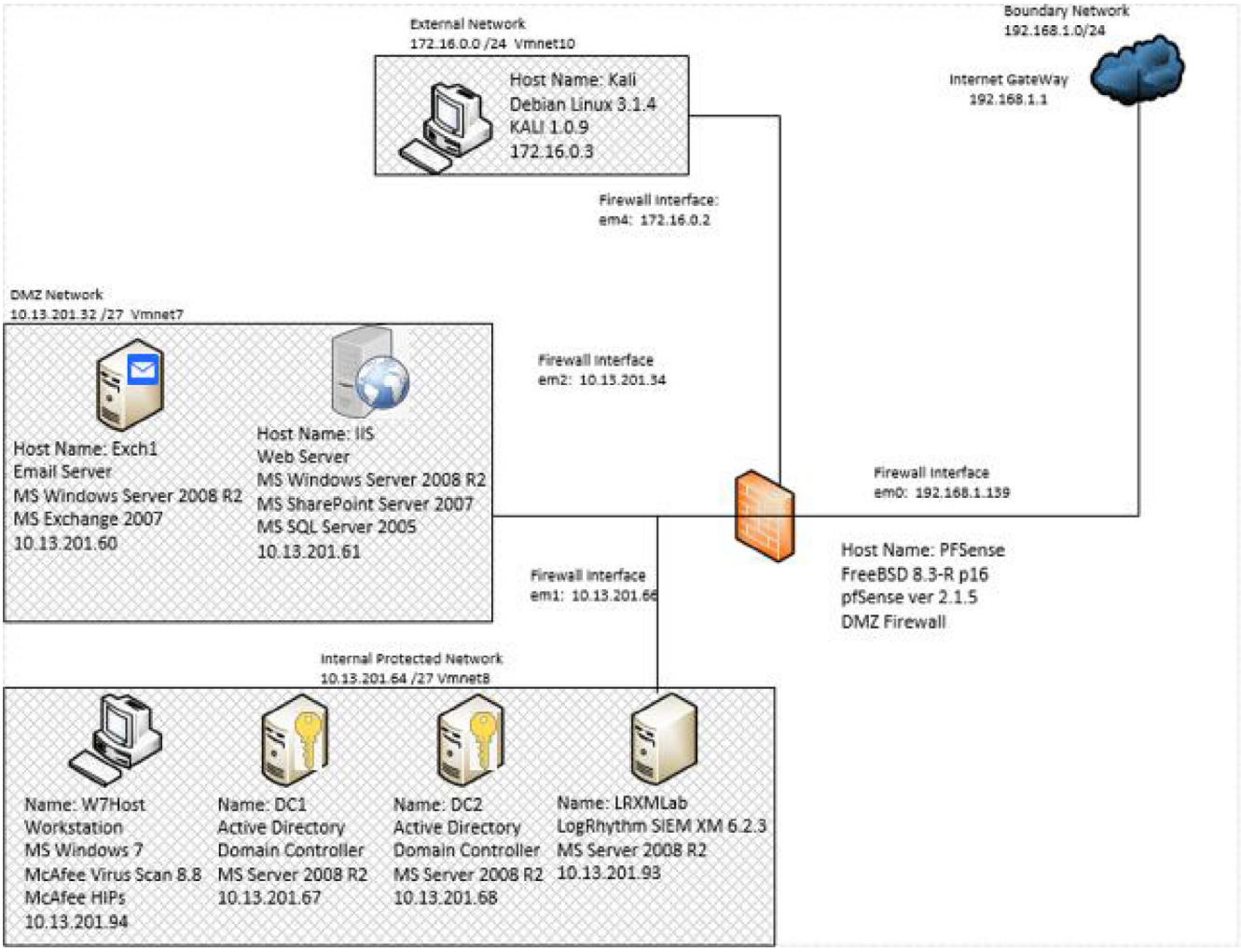


Fig. 6 – SIEM virtual network topology.

6. Future research

Research conducted during this study indicated that network security solutions exhibited differing levels of value across multiple phases. No sensor was capable of providing coverage across all attack phases; however, there were areas where multiple sensors provided partial redundancy in detections, though with varying levels of data fidelity or confidence in detection rates. It may be worth evaluating the efficacy of leveraging the Bryant Kill-Chain to assign confidence weights to security technologies within each phase in order to calculate triage priorities for analysts to reference when dealing with a high number of alarms. The Dempster-Shafer belief function has been leveraged to perform this function in past works related to alarm fusion and confidence ratings, though the reliance on a robust weighting framework was cited as a weakness of applied statistical models (Yu and Frincke, 2005). The Bryant model may overcome some of the shortcomings of the value models leveraged for statistical weighting cited in previous works.

7. Conclusions

Table 1 below summarizes the results of the evaluation in terms of raw logs generated by sensors and provided to the SIEM, alarms generated by the SIEM, and the number of normalized events contained within resultant alarms. The modified framework successfully detected 25 out of 26 test cases (96% detection rate) compared to seven out of 26 test cases (26.9% detection rate) with the default SIEM framework. It is worth noting that when both configurations detected an activity, the modified framework typically produced fewer alarms while simultaneously presenting analysts with data aggregated from a higher number of events, as was depicted in Fig. 5.

SIEM correlation rule construction with the new model was assessed to be an improvement over the default SEIM data model. The primary advantage of the new model was the ability to align log data or SIEM events with more descriptive event categories populated with real world indicators of compromise discovered by analysts using the same framework for investigations as the SIEM was leveraging for correlation and

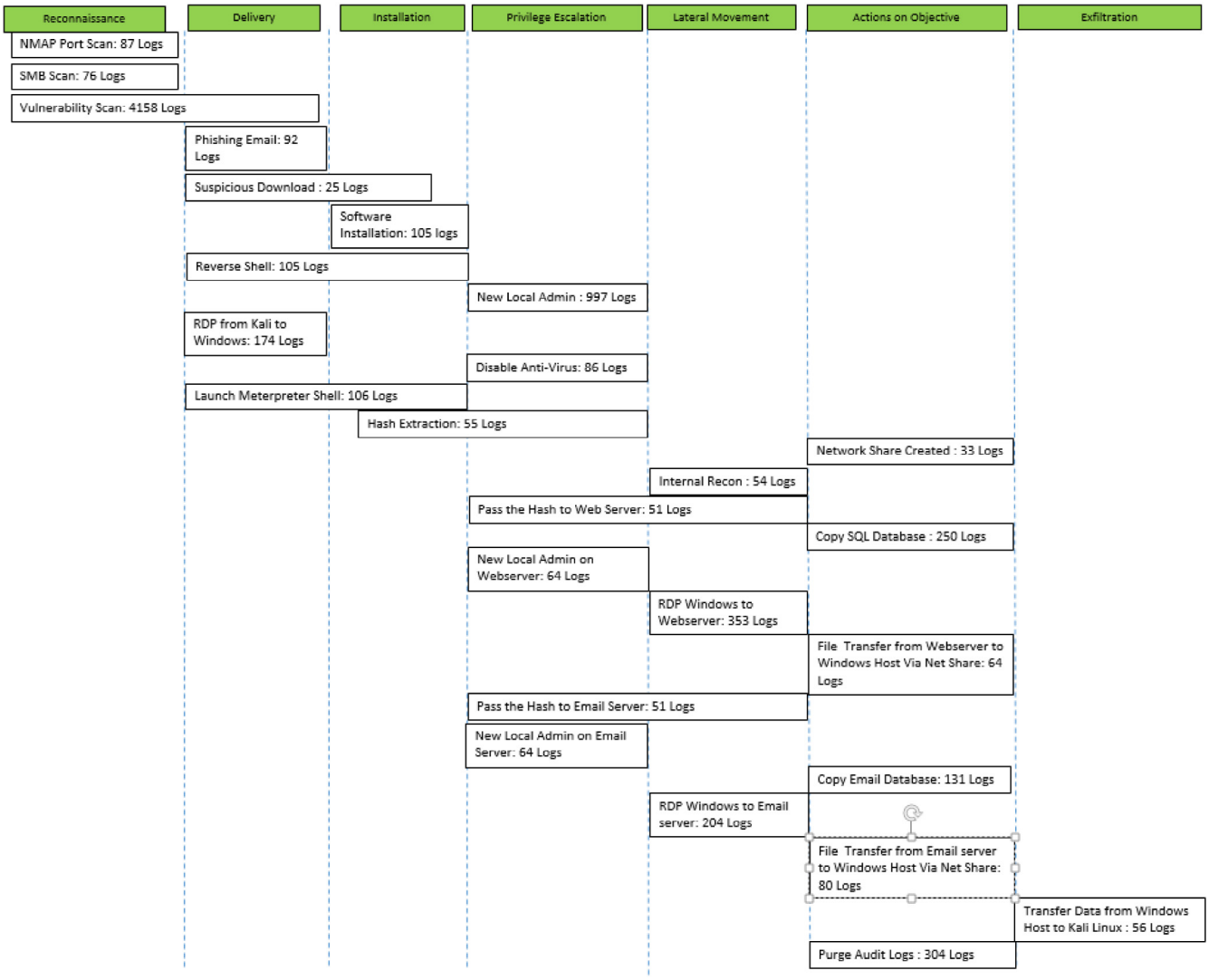


Fig. 7 – Test case and phase crosswalk: the scenario involves an attacker located on an external 172.16.x.x network attempting to access an internal 10.x.x.x network.

alarm generation. This alignment enabled alarm segregation into meaningful categories that analysts had become familiar with during postmortem data evaluation.

A secondary benefit of SIEM data alignment with the new classification model was the ability for analysts to forecast attacker actions along a spectrum of expected events. This provided context for future events the analysts expected to see following alarms, or potential areas to investigate that would have occurred prior to an alarm in order to identify root cause of the incident. Analyst feedback on SIEM alarm performance was easily communicated to SIEM engineers via the Bryant Kill-Chain as both analysts and engineers had become familiar with the same model. This provided an efficient mechanism for analysts to identify shortcomings in correlation rules when alarms were expected but not generated and communicate candidate indicators of compromise to SIEM engineers for incorporation as alarm criteria.

The Bryant Kill-Chain is assessed to be an improvement over the Lockheed Martin Kill-Chain in respect to suitability for incorporation into a SIEM system. Expanding the Lockheed Martin Kill-Chain into additional phases based on both attacker action and data similarity provided an elegant mechanism for aggregating related events into robust alarms assessed to be more valuable to security analysts.

The resultant aggregate alarms enabled analysts to perform analysis of alarm metadata without requiring post alarm queries to explain security incidents. Reducing the number of post alarm queries reduced the amount of time required for analysts to identify whether an event warranted escalation to additional security personnel. The improved detection rate, decreased overall alarm volume and additional metadata found in aggregate alarms were assessed to be improvements to the existing SIEM system and instrumental in detecting security threats in a more expeditious manner.

The improved alarm rates in both the number of detected test cases and decreased number of redundant alarms generated per test case is assessed to have improved the value of the modified SIEM system to analysts performing continuous

security monitoring. Analyst response times were assessed to have been improved due to the increased visibility resulting from these improvements in alarm quantity and quality. Metadata nesting within aggregated alarms was also assessed to improve analyst work flow due to the elimination of manual tasks associated with copying and pasting data from multiple alarms into investigation reports when escalating incidents to stakeholders.

REFERENCES

Beaver JM, Symons CT, Gillen RE. A learning system for discriminating variants of malicious network traffic, 8th Annual Cyber Security and Information Intelligence Research, ACM, pp. 1–6, 2013.

Best DM, Endert A, Kidwell D. 7 Key challenges for visualization in cyber network defense, Proceedings of the Eleventh Workshop on Visualization for Cyber Security, ACM, pp. 33–40, 2014.

Chen Y, Malin B. Detection of anomalous insiders in collaborative, Proceedings of the first ACM conference on Data and application security and privacy, ACM, pp. 63–74, 2011.

Claycomb WR, Shin D. Detecting insider activity using enhanced directory virtualization, Proceedings of the 2010 ACM workshop on Insider threats, ACM, pp. 29–36, 2010.

Flagg L, Streeter G, Potter A. Bringing knowledge to network defense, Proceedings of the 2007 Spring Simulation Multiconference, Society for Computer Simulation International, vol. 3, pp. 370–377, 2007.

Hutchins EM, Cloppert MJ, Amin RM. Intelligence-driven computer network defense informed by analysis of adversary campaigns and intrusion kill chains, Proceedings 6th International Conference Information Warfare and Security (ICIW 11), pp. 113–125, 2010.

Ioannou G, Louvieris P, Clewley N, Powell G. A Markov multi-phase transferable belief model: an application for predicting data exfiltration of APTs, 16th International Conference on Information Fusion, IEEE, pp. 842–849, 2013.

Kul G, Upadhyaya S. A preliminary cyber ontology for insider threats, Proceedings of the 7th ACM CCS International Workshop on Managing Insider Security Threats, ACM, pp. 75–78, 2015.

McWorther D. APT1: exposing one of China's cyber espionage units, February 2013. [Online]. http://intelreport.mandiant.com/Mandiant_APT1_Report.pdf. [Accessed 14 March 2017].

PWC, Key Findings from the 2015 US State of Cybecrime Survey. July 2015. [Online]. http://www.pwc.com/us/en/increasing-it-effectiveness/publications/assets/2015-us-cybercrime-survey.pdf. [Accessed 14 March 2017].

Ross R, Stoneburner G, Fabius-Greene J, Dempsey K, Bodeau D, Caddy C, et al. Managing information security risk: organisation, mission, and information system view. National Institute of Standards and Technology; 2011.

Rush G, Tauritz DR, Kent AD. Coevolutionary agent-based network defense lightweight, Proceedings of the Companion Publication of the 2015 Annual Conference on Genetic and Evolutionary Computation, ACM, pp. 859–866, 2015.

Shalyapin A, Zhukov V. Case based analysis in information security incidents, Proceedings of the 8th International Conference on Security of Information and Networks, ACM, pp. 312–317, 2015.

Shiva S, Roy S, Dasgupta D. Game theory for cyber security, Proceedings of the Sixth Annual Workshop on Cyber Security and Information Research, ACM, 2010. doi:10.1145/1852666.1852704 [Accessed 14 March 2017].

Yu D, Frincke D. Alert confidence fusion in intrusion detection systems with extended Dempster-Shafer theory, Proceedings of the 43rd Annual Southeast Regional Conference, ACM, vol. 2, pp. 142–147, 2005.

Hossein Saiedian (Ph.D., IEEE PSEM, Kansas State University, 1989) is currently an associate chair, the director of IT degree programs, and a professor of computing and information technology at the Department of Electrical Engineering and Computer Science at the University of Kansas (KU) and a member of the KU Information and Telecommunication Technology Center (ITTC). Professor Saiedian has over 160 publications in a variety of topics in software engineering, computer science, information security, and information technology. His research in the past has been supported by the NSF as well as other national and regional foundations.

Blake D. Bryant, MS, CISSP, MCITP, CCNA, completed his MS degree in information technology at the University of Kansas in spring of 2016 and plans to continue with graduate studies to obtain his Ph.D. degree. He is a security professional with 10+ of experience in leading computer security organizations (private and military) and is currently with Ernst & Young. Bryant is also a member of the US Army Reserves.