Computers & Security 94 (2020) 101817

Contents lists available at ScienceDirect

Computers & Security

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

Improving SIEM alert metadata aggregation with a novel kill-chain based classification model

Blake D. Bryant, Hossein Saiedian*

Electrical Engineering & Computer Science, The University of Kansas, Lawrence, KS 66049, USA

ARTICLE INFO

Article history:
Received 7 November 2019
Revised 16 March 2020
Accepted 23 March 2020
Available online 4 April 2020

Keywords:
Network monitoring
Intrusion detection
Kill-chain
Advanced persistent threat
APT
Security information and event management
SIEM
Security log ontology
Computer network defense
Attack ontology
Threat framework

ABSTRACT

Today's information networks face increasingly sophisticated and persistent threats, where new threat tools and vulnerability exploits often outpace advancements in intrusion detection systems. Current detection systems often create too many alerts, which contain insufficient data for analysts. As a result, the vast majority of alerts are ignored, contributing to security breaches that might otherwise have been prevented. Security Information and Event Management (SIEM) software is a recent development designed to improve alert volume and content by correlating data from multiple sensors. However, insufficient SIEM configuration has thus far limited the promise of SIEM software for improving intrusion detection. The focus of our research is the implementation of a hybrid kill-chain framework as a novel configuration of SIEM software. Our research resulted in a new log ontology capable of normalizing security sensor data in accordance with modern threat research. New SIEM correlation rules were developed using the new log ontology, and the effectiveness of the new configuration was tested against a baseline configuration. The novel configuration was shown to improve detection rates, give more descriptive alerts, and lower the number of false positive alerts.

© 2020 Elsevier Ltd. All rights reserved.

1. Introduction

The goal of any network security monitoring solution is timely, accurate and actionable network threat alerts. Such alerts are cited as an axiom of mature security organizations such as the U.S. Department of Homeland Security Grano et al. (2005). Unfortunately, security alerts are often prone to false positives based on sensor location within the network, limitations in their ability to apply advanced rule logic, or the inability to represent complex organizational data hierarchies such as: user accounts, critical computing resources, subnet risk levels, and work hours. Additionally, individual security devices themselves may be susceptible to exploitation by savvy attackers, affecting the integrity of data they provide Garcia et al. (2018). These limitations result in a multitude of alerts flooding security analysts, or a lack of alerts due to overzealous alert suppression.

Recently, leading information security companies have developed specialized correlation software designed to aggregate data provided by disparate sensor feeds, thus enabling holistic analysis of all network data from a single, centralized, alert feed. Analysis of data from these devices may reveal patterns of activity conducive to fingerprinting individuals or threat groups, based on a trail of data spread across an entire network of sensors. However, it is the development of custom algorithms designed to analyze this data in the context of phased attack ontologies that truly provide additive value in threat detection and prevention.

Additionally, there is a trend of chatty data feeds, such as firewall logs, drastically outnumbering more forensically valuable data feeds such as endpoint operating system logs. Observation of security analysts operating in a commercial Security Operations Center indicated that a vast majority of security alerts were ignored by analysts. Furthermore, many security experts argue that weeding through every alarm is impractical and often must be combined with some form of automation for attack attribution Aminanto et al. (2019), Zhong et al. (2016) and Zhong et al. (2019). Unfortunately, merely aggregating data from sensors does not greatly improve detection rates nor decrease false-positive ratios.

Discerning notable security events from log data, and implementing timely remediation for incidents, is a daunting task without an effective alerting engine employed to filter, categorize and escalate security events appropriately. Security data must be normalized into a standard ontological framework, analyzed within the context of known attacker methodologies, and finally allowed

* Corresponding author.
E-mail address: saiedian@ku.edu (H. Saiedian).

https://doi.org/10.1016/j.cose.2020.101817
0167-4048/© 2020 Elsevier Ltd. All rights reserved.

to accrue suspicion dynamically as threat activity progresses throughout the network to fully realize the axiom of timely, accurate and actionable alerts.

1.1. Significance

Advanced correlation software in SIEM systems is designed for real-time alerting of potential security events, as well as to increase the investigative and data retrieval functions associated with those events. Analysis of raw sensor feeds is overwhelming for human analysts due to the high volume of alerts and high false positive ratios. Some studies have revealed as few as 29% of alerts in a SOC environment are actually inspected by analysts, of which an average of 40% are determined to be false positives Zhong et al. (2019). Implementing programmatic analysis decreases false positive ratios and provides mechanisms for the abstraction of human labor functions to a higher analytical plane via a unified graphical user inter-face (GUI). This in turn enables the establishment of analyst pools ultimately improving process efficiency and decreasing the mean time required to triage and respond to net-work security events.

However, current software solutions for data normalization and threat action modeling within SIEM software are limited. These solutions merely provide a framework for normalizing disparate data feeds and performing logical comparisons of the metadata contained therein. Such tools are often used to implement static trigger criteria based on either volumetric thresholds or watch lists containing threat signatures, but this methodology is prone to false detection.

A method of implementing dynamic suspicion escalation through contextualized data, aggregated from multiple sources, and attributable to specific threat actions is not found within SIEM software by default. A threat framework must first be adopted to attribute malicious activity to specific threat objectives. This framework can then be leveraged to attribute various levels of risk and suspicion according to the extent to which activities satisfy the threat objective phases.

This paper analyzes existing threat frameworks for inclusion within a SIEM solution, with the goal of providing more timely, accurate and actionable alerts through threat attribution and dynamic suspicion escalation. Ultimately, a novel threat model was devised based on the competing threat models evaluated. This novel model was implemented through modifications to the database structure of a commercially available SIEM system.

1.2. Research methodology

An empirical research methodology was applied to evaluate existing research associated with intrusion detection technology, SIEM software, and network attack methodologies. This study was conducted over a period of two years and included immersion within a commercial Security Operations Center (SOC) to observe security analysts conducting alert triage and investigation during real-world security incidents. The concepts of data triage, suspicion escalation, threat actor groups, and models for representing threat methodologies were evaluated within this environment. This study led to the selection of a commercial SIEM product for evaluating the efficacy of implementing ontological frameworks used to represent security data in a normalized format. The LogRhythm SIEM was chosen as it was the dominant SIEM system leveraged by the security analysts during the observation period. Finally, a laboratory environment was constructed to validate insights gained from observing SOC analysts by implementing the newly devised SIEM framework within a controlled environment. The laboratory environment consisted of a security device sensor array, multiple security devices configured in series, and the selected SIEM product. Fig. 1 illustrates this laboratory design. Fig. 14, within the testing and evaluation section, illustrates the network architecture and provides context for network traffic flow.

SIEM correlation rules were implemented in accordance with the model devised in this paper. Detection performance was evaluated in relation to a baseline SIEM configuration with vendor recommended correlation rules.

2. Background

2.1. Hacker categories

Research on network intrusion detection systems, and security event management systems have existed for several years; however, most research focus on technical challenges associated with data analysis rather than psychological motivations of attackers Denning (2001),Denning (1987),Garcia-Teodoro et al. (2009) and Valeur et al. (2004). An ontological framework representing how persistent threat groups penetrate networks and exploit vulnerabilities is seldom addressed in contemporary research on intrusion detection.

However, some work has been done to understand the human factors associated with cyber criminals. Hald and Pedersen categorized threat actors based on expertise and motivation, as depicted in Fig. 2 (Hald and Pedersen, 2012). Such work associated with the attribution of actions to specific types of threat actors is an important step in eventually establishing a pattern of behavior conducive to data correlation and attribution of malfeasance.

This paper focuses primarily on the Information Warrior (IW) category described by Hald and Pedersen. As such, though the security landscape is filled with commodity malware or drive-by exploits, such events are not the focus of this study. The intent of this paper is not to discount the threat of abbreviated cyber attacks, but rather focus on the more challenging multi-stage incidents that require greater analytical rigor by security analysts. Therefore, this paper focuses on the well resourced, focused and persistent actions typically associated with organized threat groups.

2.2. The attack lifecycle: kill-chains

The term “kill-chain” emerged in security circles in the early2010’s as a way to describe the lifecycle of a security incident. The term kill-chain is derived from the Department of Defense joint targeting process, which was designed for the positive identification and attribution of culpability to actions associated with suspected actors. The US targeting kill-chain is epitomized by the acronym F2T2EA, which consists of six phases: Find, Fix, Track, Target, Engage and Assess. This is similar to a pipe and filter model in software engineering, with the product of one phase providing input to subsequent phases in a serial fashion. Disruption of any phase within this chain will result in the dissolution of the process in its entirety. The following studies outline kill-chain model research.

The Lockheed martin intrusion kill-chain The most prominent model associated with the term “kill-chain” within network security research is the Lockheed Martin Intrusion Kill-Chain depicted in Fig. 3. In this model, Advanced Persistent Threats (APTs) employ a methodical targeting process similar to the DoD kill-chain (Eric Hutchins and Amin, 2011). Lockheed Martin’s “intrusion kill-chain” describes the seven phases of activities APTs conduct to compromise a system: Reconnaissance, Weaponization, Delivery, Exploitation, Installation, Command and Control, and Actions on the Objective. However, the Lockheed Martin model does not adequately address actions other than data exfiltration which can occur after a persistent threat has compromised a system. For

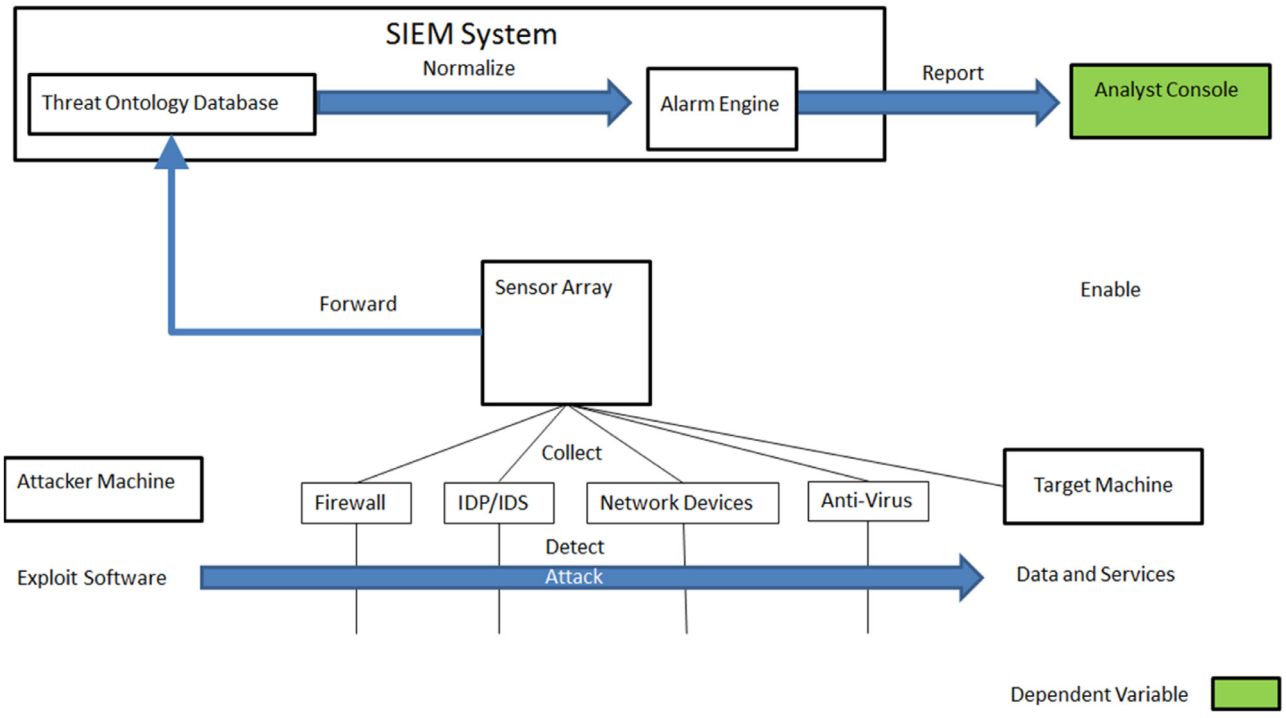

Fig. 1. Laboratory concept configuration depicting an array of multiple security sensors reporting to a SIEM system Figure 16.

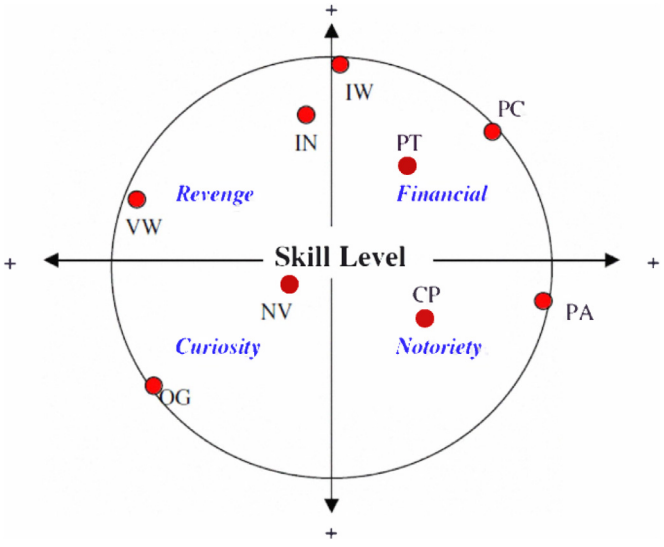

Fig. 2. The Hald and Pedersen motivation/skill-level circumplex Hald and Pedersen (2012) depicts varying skill levels associated with categories of hackers.

instance, lateral reconnaissance to determine more susceptible systems, followed by repetition of phases one through six, is not addressed by this model. It is uncommon for an advanced threat to attempt to transfer data from the initial compromised system, since such activity increases suspicion and may risk loss of the system as a persistent access point into the network.

Mandiant APT attack lifecycle model The Mandiant Corporation devised an eight phase model, depicted in Fig. 4, called the "Mandiant APT Attack Lifecycle," which includes the iterative process attackers employ to gain additional footholds within a network following the initial compromise (FireEye, 2013). This model considers the possibility of branch and recursion at phase five, spawning sub-phases associated with lateral infection. The Mandiant model greatly simplifies the initial phases of the Lockheed Martin killchain by incorporating the weaponization, delivery, exploitation, and installation phases into a single phase called initial compromise. The Mandiant model also labels phases based upon intent rather than action, helping to aggregate actions that serve a common purpose. Another key differentiator between the Lockheed Martin model and the Mandiant model is the escalate privileges phase. Mandiant identifies multiple tools used by APT groups to gain access to additional resources on the compromised system, which provide behavioral signatures that may serve as key indicators of compromise and differentiate between routine and persistent threat activity.

Both the Lockheed Martin and Mandiant frameworks inspired security analysts to provide methodical approaches to security data triage within the context of an attacker's perspective. Additionally, these frameworks focused on consolidating threat activities into discrete groups based on attacker objectives, rather than exhaustive lists of tools, techniques or signatures observed in the wild. The ability to organize threat activity in this manner

Fig. 3. The Lockheed-Martin Killchain Eric Hutchins and Amin (2011) depicts a seven phase attack lifecycle.

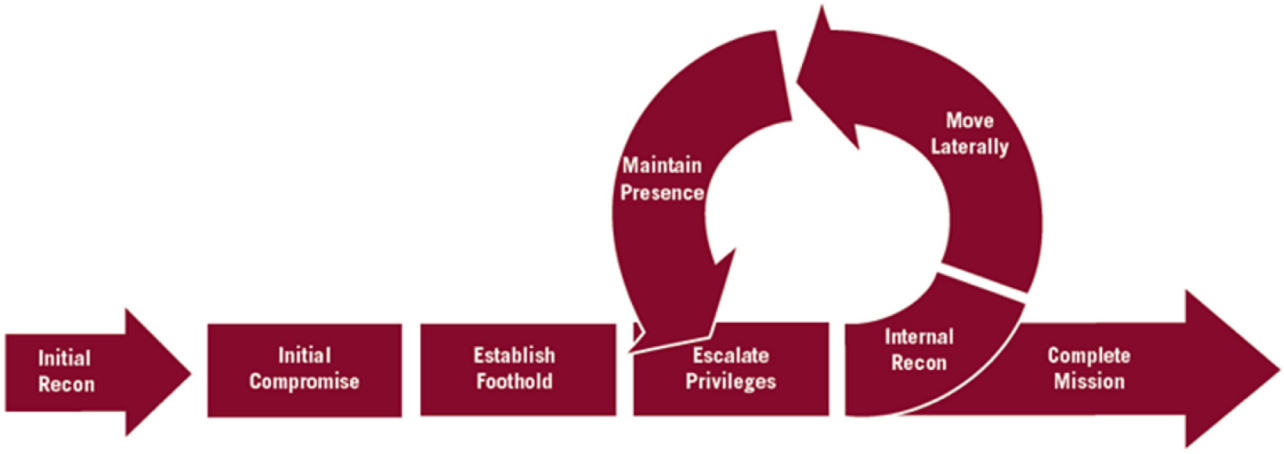

Fig. 4. Mandiant Attack Lifecycle FireEye (2013) depicts an eight phase attack lifecycle with possible recursion.

Initial Access	Execution	Persistence	Privilege Escalation
Drive-by Compromise	AppleScript	.bash_profile and .bashrc	Access Token Manipulation
Exploit Public-Facing Application			
External Remote Services			
Hardware Additions			
Replication Through Removable Media			
Spearphishing Attachment			
Spearphishing Link			
Spearphishing via Service			
Supply Chain Compromise			
Trusted Relationship			
Valid Accounts			

Defense Evasion	Credential Access	Discovery	Lateral Movement
Access Token Manipulation	Account Manipulation	Account Discovery	AppleScript
Binary Padding			
BITS Jobs			
Bypass User Account Control			
Clear Command History			
CMSTP			
Code Signing			
Compile After Delivery			
Compiled HTML File			
Component Firmware			
Component Object Model Hijacking			

Collection	Command and Control	Exfiltration	Impact
Audio Capture	Commonly Used Port	Automated Exfiltration	Account Access Removal
Automated Collection	Communication Through Removable Media	Data Compressed	Data Destruction
Clipboard Data	Connection Proxy	Data Encrypted	Data Encrypted for Impact
Data from Information Repositories	Custom Command and Control Protocol	Data Transfer Size Limits	Defacement
Data from Local System	Custom Cryptographic Protocol	Exfiltration Over Alternative Protocol	Disk Content Wipe
Data from Network Shared Drive	Data Encoding	Exfiltration Over Command and Control Channel	Disk Structure Wipe
Data from Removable Media	Data Obfuscation	Exfiltration Over Other Network Medium	Endpoint Denial of Service
Data Staged	Domain Fronting	Exfiltration Over Physical Medium	Firmware Corruption
Email Collection	Domain Generation Algorithms	Scheduled Transfer	Inhibit System Recovery
Input Capture	Fallback Channels		Network Denial of Service
Man in the Browser	Multi-hop Proxy		Resource Hijacking

Fig. 5. The MITRE ATT&CK™ framework MITRE (2014) depicts a twelve phase model and provides references to techniques used in each phase.

provides a mechanism for methodical triage of security data, rather than weeding through a sea of disjointed security alarms.

2.3. Hacker methods

The Mitre organization developed the ATT&CK™ framework in 2018, depicted in Fig. 5, as a way to categorize observed threat behavior across a twelve phase model along with observed threat techniques associated with each phase Strom et al. (2018). This model is the logical evolution of applying the attack lifecycle concepts championed by the Lockheed Martin and Mandiant models with the traditional approach of developing attack signatures. This framework is available on the Mitre organization's website and is updated as new adversary tactics and techniques are identified.

2.4. Security information and event management (SIEM) software

Amrit Williams and Mark Nicolett coined the term SIEM in 2005 to describe the convergence of Security Event Management (SEM) and Security Information Management (SIM) software into a single consolidated product Williams and Nicolett (2005). Historically, SIM software was focused on post-incident review and

analytics, while SEM software was designed to provide real-time alerting of intrusions or other security incidents. SIEM products additionally provide log management services since log collection, analysis, and retention are integral parts of the SIEM process.

Several papers have been written to address individual components that provide data for SIEM systems, such as improving detection ratios in low level sensors Kim et al. (2013), log retention and management data structures Madani et al. (2011), and packet inspection Silowash et al. (2013). However, few studies have been conducted on SIEM software and its underlying mechanisms: security event management, threat taxonomies, attack ontologies, and incident weighting. Understanding these mechanisms provides insight to potential areas for optimization.

SEM systems focus on the process of actively detecting security events as they occur. The following SEM models established the theoretical basis for future SIEM systems.

2.5. Progression of dangerousness

Legrand addressed the task of wading through holistic network analysis alerts by subjecting normalized SEM data to a static causal event ontology based on five factors: why, who, where, how and what Legrand et al. (2008). Each ontological factor must be satisfied by an observable network event and the summation of these events constitutes an action. The result of this ontological analysis is run through a threat algorithm called the progression of dangerousness, where actions are weighted to identify which are the most threatening to network assets. Action weighting is calculated via the function

$$f(a) = (d_1(a), d_2(a), \cdots, d_p(a))$$

where each observable action a is iteratively evaluated against all ontological dimensions d_1 through d_p. This model relies heavily on the intrinsic detection capabilities of sensors.

Chien et al. proposed a two-layer attack framework where primitive attack (PA) sensor information feeds into an attack subplan layer, based upon attack subontology and attacker intent Chien et al. (2007). The ontology has three classes: reconnaissance, penetration, and unauthorized activity. This signifies the transition from static ontological analysis to a dynamic ontology with classes dependent upon the state transitions between PAs. Chien also introduced the notion of assigning confidence values to detection on a per sensor basis. Chien's primitive attack layer expands upon other event verification module concepts as well as incorporating Legrand's concept of ontological integration. Higher level subplan templates are used to align disparate PA information into a coherent attack based on known or suspected attack methodologies.

Visualization and graphical tools SIEM software may be improved with visualization tools for postmortem incident auditing and predictive analysis. Kotenko and Novikova outlined the essential functions of a SIEM visualization subsystem: Real time data monitoring, integration with a historical data repository, graphical interface for rule editing and generation, attack modeling, and resource management Novikova and Kotenko (2013). Histograms, linear diagrams, and dashboards are all useful.

Filtering and Correlation Flynn focused on implementing kill-chain methodologies in SIEM software and stressed collecting event data on routine activity so holistic analysis may be conducted on security incidents Flynn (2012). A continuum of progressive suspicion is needed, similar to Legrand's progression of dangerousness. Flynn proposed an "event pipeline" framework consisting of blacklisting, identity translation, correlation, context, and analysis. Blacklisting in this context is the removal of known false positives, such as those which match signatures stored on intrusion detection systems, but which are associated with operating systems that are not in the network. Identity translation entails maintaining a record of internal machines, users, and IP addresses for future correlation. Correlation has two sub-phases: the attack plane and the kill-chain Flynn (2012). The attack plane compares disparate events with some shared identifying characteristics to determine group events for context and suspicion escalation. The Lockheed Martin model is the basis for the kill-chain, which provides criteria for attack plane grouping. Context is the fusing of external information surrounding the detection, such as cross-referencing network diagrams. In Analysis, a correlated and contextualized alert is provided to a human for review.

3. Security operations center study and observations

3.1. SOC environment overview

The authors of this paper were provided with unfettered access to a leading Managed Security Services Provider's (MSSP) Security Operations Center (SOC) over a two year period. Data processed within the SOC was associated with a growing list of clientele reaching more than 120 distinct clients across the globe by the end of the study. Client profiles extended across multiple industry verticals including: retail, health services, gambling, utilities, education, hotels, and the public sector.

The MSSP employed a total of 22 SOC security analysts during the study period. Analysts were aligned within a three-tiered model based on analyst experience and rigor of expected investigative effort. Tier 1 analysts were responsible for alert triage and escalation of routine incidents to clients. Tier 2 analysts were responsible for handling internal escalations for abnormal activity or validation of suspected false positive events relayed by tier 1 analysts. Tier 3 analysts were responsible for in depth investigations and response actions associated with known or suspected security breaches or client initiated investigations.

SOC analysts were responsible for responding to alerts and performing investigations within several different SIEM systems including: McAfee ESM (formerly Nitro Security), IBM QRadar, ArcSight, Splunk and LogRhythm. Alarms were provided to analysts either via a remote console into the SIEM management system or via email alerts automatically generated by the SIEM system. The LogRhythm SIEM system was configured as a multitenant system servicing the majority of the MSSP clients simultaneously via a cloud-based deployment.

The multitenant LogRhythm deployment provided a consistent basic SIEM correlation rule set across multiple clients, with the option for select clients to request additional rules above the base configuration. Single client SIEM deployments varied greatly from one another in terms of correlation rule construction, data source integration and possibly metadata parsing standards. Single client SIEM systems were not a primary focus during this study due to the large variance in system configuration and relative complexity in accessing systems vice the convenience of a multitenant console. Therefore, the SOC study focused primarily on data collected within the LogRhythm SIEM as it provided direct access to SIEM data via a management console, allowed for real-time log queries, and contained the largest variety of client data across multiple industry verticals.

The MSSP also employed a total of five SIEM engineers, also aligned within a three-tiered model. SIEM engineers provided support to each of the SIEMs analysts operated within. SIEM engineers were expected to be experts in at least one SIEM system, but also possess working knowledge of all other systems. The authors were allowed to provide recommendations for potential correlation rules to SIEM engineers during the study period, presenting the opportunity to review existing and pending correlation rule construction. Observation of SIEM correlation rule construction provided insight

into the relative complexity of correlation as well as best practices for rule development.

Security data processed by SIEM systems and analysts grew at an exponential rate as clients flocked to the MSSP. The multitenant SIEM serviced 12 clients at the beginning of the study and grew to more than 60 by the end of the two-year period. By the end of the study, several million logs were processed by the LogRhythm SIEM on a daily basis resulting in over 4000 alarms for analysts to triage, investigate and/or escalate to clients at the peak of the study. Analysts were expected to conduct initial triage and client notification (if warranted) with a 15-minute service level agreement.

3.2. Security analyst overview

Analysts operated across five different shifts in order to provide 24x7x365 coverage, were expected to work on holidays, but were offered additional vacation days to make up for holiday assignments. The weekday day shift operated from 0800 to 1600h on Mondays through Fridays. Monday through Thursday were augmented by a "swing" shift from 1200 to 2200 h, and an evening shift from 2100 to 0900 hrs. The weekend shifts worked Friday morning to Monday morning from 0800 to 2100 and 2000–0900.

Symptoms of analyst burnout and alert fatigue were most prevalent during the weekday day shift from 0800-1600hrs. The weekday day shift was particularly problematic for security analysts as they were responsible for addressing client inquiries via email or phone while also responding to SIEM generated alerts. Client inquiries typically tapered off by 1600 client local time, except for Friday afternoons, which exhibited a large increase in client inquiries between 1500–1700 client local time. The large increase in Friday client inquiries is suspected to be associated with client attempts to investigate incidents during the week without MSSP involvement and escalating unresolved issues to the MSSP prior to retiring for the weekend.

Analyst interactions with clients proved to be a very time-consuming process. Typical email exchanges between clients and analysts took an average of five minutes to draft per message. Phone calls between analysts and clients typically took longer than email correspondence and eventually became burdensome to the point where select clients were scheduled for prearranged 30-min calls with dedicated analysts on a weekly basis. Eventually the MSSP created a new section responsible for handling interactions with clients in an effort to alleviate the pressure placed on analysts. However, analysts were often required to continue to attend client calls based on a lack of security expertise on the part of the newly appointed client services team.

Analysts used the US CERT Federal Agency Incident Categories in order to standardize incident reporting to clients. The US CERT categories are based on a seven level zero based system, with lower categories representing more pressing issues. Category 0 is reserved for known security tests and was omitted from reporting requirements by analysts. However, blind penetration tests conducted by clients were expected to be reported by analysts and would have been considered a category 1 event. Category 1 was reserved from a suspected security breach resulting in unauthorized access. Category 2 was reserved from successful denial of service attacks. Category 3 was reserved from malware or other malicious code detection. Category 4 was reserved for "improper usage" and was seldom defined or implemented in client environments. Category 5 was reserved for scans, probes or attempted but unsuccessful access attempts. Finally, category 6 was reserved from unconfirmed incidents and served as a "catch all" for alerts analysts chose not to escalate to clients.

Every alert in the multitenant SIEM was expected to be manually assigned one of the 7 US CERT categories by an analyst. More severe alert categories (CAT1-CAT3) required immediate escalation and a phone call to a client. Category 5 and 6 alerts only required email notifications or consolidation into daily or weekly reports based on client preference. The vast majority of alerts were assigned to the "category 6 catch all" category. The analyst triage process typically began by reviewing the name of alarms produced by the SIEM. Well named alarms provided analysts with insight pertaining to which actions should be conducted during triage. Poorly named or vague alerts were typically ignored by analysts, unless they were occurring very frequently, in which case they were escalated to SIEM engineers for "tuning" or removal from the system.

Furthermore, alarms that could not be easily explained by analysts were often not sent to clients. The was likely due to client pushback after receiving several escalations from analysts that were interpreted as being unactionable. The most common alarms that analysts received pushback from clients for were associated with IP blacklists, wherein one or more systems were observed communicating with a system previously known for malicious activity. These alerts were typically triggered off of network data, with little context, and did not provided insight as to the nature of the offense. Clients quickly developed a policy of inquiring for more data following escalations, such as account names involved, processes running on target systems, and actions performed by the attacker, all of which required additional resources from analysts to collect. Ultimately, escalation rates for alarms were directly proportional to analysts' ability to rapidly identify malicious activity and collect adequate forensic evidence to make remediation recommendations to clients. If the evidence required for the latter was lacking, analysts opted not to escalate alarms.

3.3. SIEM engineer overview

SIEM engineers operated predominantly during the weekday day shift hours and maintained an on-call roster for after-hours emergencies. SIEM engineers occasionally scheduled work during evenings or weekends if prearranged maintenance windows were requested by the client for large installations or major system modifications. SIEM engineers operated off of a ticket based workflow wherein configuration requests could be initiated by either clients or SOC analysts. Client requests typically involved the configuration of sensor data feeds, managing alert thresholds (also known as tuning), and custom SIEM alert rule development. Analyst requests typical involved alert threshold configuration requests to silence "chatty" or "bad" SIEM alert rules. SIEM alerts that were evaluated as "bad" by tier 1 or 2 analysts were required to be reviewed by a tier 3 analyst before being escalated to SIEM engineers for resolution. Tier 3 analyst review of alarm quality was required to prevent lower tier analysts from merely removing alerts they did not wish to report from the system.

SIEM engineers were expected to interact with clients on a regular basis in order to maintain customer satisfaction with the services provided. During initial client integration, SIEM engineers conducted calls or video meetings with clients several times a week to install or configure security data streams. Data streams were monitored for an approximately two-week period before setting baselines or alarm thresholds. After this learning period, alerts were enabled and began flooding SOC analyst consoles. Excessively noisy alerts were reported by SOC analysts to SIEM engineers as candidates for tuning. This tuning process typically lasted for two weeks following the initial log baselining period.

The variety of security data sources evolved as the study progressed. During the initial phases of the study, clients primarily forwarded firewall and IDS alerts to the SIEM but did not forward operating system audit logs or specialized security devices (aside from network based IDS). A few select clients chose to forward network device logs from routers and switches. Eventually clients

began integrating more exotic data sources into the SIEM environment. This was partially motivated by media reports of high-profile security breaches, but also the adoption of larger companies as clients with more mature security organizations. Eventually monitored data sources expanded to include operating system logs, specialized anti-malware solutions, host-based IDS, web proxies, and vulnerability scanners. Developing parsing rules for this large variety of heterogeneous data sources eventually posed the most significant challenge to SIEM engineers; especially due to the fact that no agreed upon standard existed for metadata normalization, and not every SIEM normalized data in the same manner. Though most system vendors, such as firewall manufacturers, provided similar data for like systems, each did so with proprietary labels that had to be interpreted by the SIEM into a common representation for correlation.

The SIEM used to automatically triage log data and generate alarms implemented a system of event classification labels in order to prioritize observed traffic. All log data ingested by the SIEM received a classification label during the initial data normalization phase, wherein metadata was aligned with standardized fields for correlation across disparate data streams. Normalized log data could also be used to create "events" stored in the LogRhythm database. Events were similar to logs, in that they could be retrieved during investigations and could be used to generate alerts; however, events, unlike logs, could be generated off of one or more logs or events using if-this-then-that logical operations. Additionally, newly generated events could be labeled with classification categories or event titles different from the labels associated with the logs or events used to generate them.

By default, the LogRhythm SIEM was configured with three top level classifications "security", "operations" and "audit". During this study, SIEM alarms were predominantly created to trigger from logs with the security or operations classifications, while audit logs were primarily used for generating daily activity reports. Fig. 6 depicts the default classification labels available to SIEM engineers using the LogRhythm SIEM.

SIEM correlation rules could be used to generate alarms, which would be sent to the analyst consoles for triage, or via email notification, or both. Additionally, correlation rules could be constructed to convert observed logs or events into additional events. Events could be queried from the SIEM database during investigations, used to generate future alerts and events, or included in reports as desired.

The practice of generating non-alerting events was not heavily used by SIEM engineers during the SOC study. SIEM engineers were typically told by clients or analysts to generate alert notifications for specific events or the presence of well-known signatures within log data. SIEM engineers employed multiple approaches to SIEM rule construction and attended regular vendor training sessions to discuss rule construction techniques. The different approaches SIEM engineers used toward rule construction are explained in the following paragraphs.

Single block rules These rules were the simplest to construct and the most likely to generate alarms on a consistent basis. This type of rule simply involved querying the SIEM log or event databases for specific elements of normalized metadata and generating an alert. The most common method for creating these rules was by comparing metadata fields with lists of known indicators. For example, lists of IP addresses associated with known malware or threat groups could be used to generate alerts whenever said IP addresses were observed within log or event metadata fields.

Multi block rules These rules were constructed using multiple if-this-then-that conditional statements executed in series. These rules were intended to perform automated triage and potentially provide additional context for alerts generated. However, there were several limitations to this approach that eventually caused it to fall out of favor with SIEM engineers.

First, this approach required several conditions to be met in order to trigger an alert. The likelihood that all conditions would be met decreased with each subsequent rule block. Originally, this phenomenon appeared to work as intended, as too many alerts overwhelm analysts. However, there was no mechanism for informing analysts or SIEM engineers that some, or most of the alerting criteria had been satisfied, but an alert was not triggered because of a technical oversight or faulty configuration setting. The end result was that this type of SIEM rule was highly prone to false negatives, wherein an alarm should have been generated, but was not.

The second issue with this approach was that it often resulted in poor forensic value to analysts. This was due to the way the LogRhythm SIEM retrieved data from its log and event datastores when generating alerts. SIEM engineers were required to select at least one metadata field as the "primary" field when generating correlation rules. The primary field was used to satisfy conditions for the rule. Engineers could optionally select additional "group by" meta data fields which would be included as populated meta data fields in the newly generated event or alert if the primary condition was met and the "group by" meta data field was not null. If a rule was configured with a specific "group by" meta data field, and a log or event met the primary criteria for the rule block, but did not contain data in one or more "group by" fields, the rule block would fail to fire for that log or event. Initially, SIEM engineers attempted to include as many "group by" fields as possible to satisfy security analyst and client demands for more detailed alarms; however, overzealous attempts to populate alarms with additional meta data resulted in most of these alarms failing to trigger. As such, this approach to SIEM rule construction resulted in rules either containing very few additional "group by" fields or being restricted to specific data sources with known consistent meta data fields to query.

Statistical rules These rules were intended to implement anomaly detection functionality within the SIEM. All of these rules operated in a multi block fashion consisting of two rule blocks, with the first block establishing the primary criteria for observation and a learning period from which a baseline could be established. The second block was used to establish the threshold beyond the baseline for triggering an alarm.

This type of alarm suffered from the same limitations as the multi block rules, in that few, if any, meta data fields would be returned within alerts. However, this rule also introduced an additional performance penalty associated with storing statistical data in memory for continual comparison with real-time data. This performance tax eventually resulted in this type of rule being reserved for only select use cases. Despite its limitations, this type of rule could be beneficial in generating suspicious events for inclusion in multi stage rules, as the title of events generated by this approach could be included in a multi stage rule and provide context, even if most or all meta data fields were omitted in the statistical event.

Multistage rules These rules were rarely used by SIEM engineers but offered the best tradeoff between forensic value and performance impact on the SIEM. Multistage rules relied on the ability of the SIEM to generate events for suspicious behavior, leverage suspicious events as alert criteria, and then eventually aggregate said events into single alerts containing data from each event observed. These rules operated in a similar fashion to multi block rules, as multiple criteria were required to be satisfied prior to generating an alert. However, unlike multi block rules, multistage rules were not limited to the construction of rule blocks configured in series within a single correlation rule. Rather, multistage rules relied heavily on the ability of a SIEM to use conditional statements to generate events and apply appropriate classification labels to said

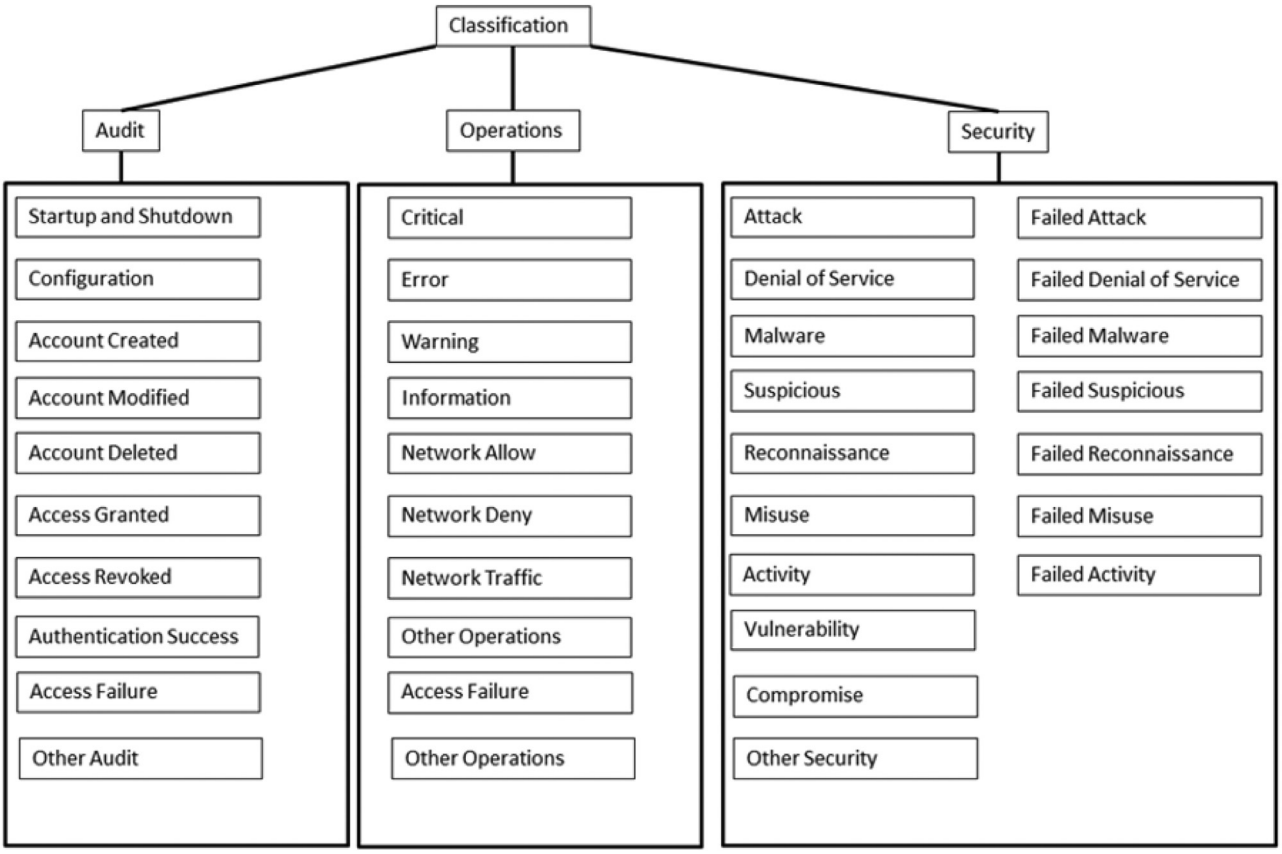

Fig. 6. LogRhythm SIEM log and event classification labels.

events. Once suspicious events were generated in the first stage, their classification label could be used as the primary criteria for generating another event or alarm in the second or subsequent stages of the series of rules. Since the first stage in this approach was merely intended to create a new event with the appropriate classification label and a descriptive title, there was little risk in applying too many "group by" fields for extracting meta data from observed logs or events as there was no second rule block that could fail. Additionally, any data source that was incapable of providing adequate meta data would be omitted from future correlation stages, thus preventing SIEM rules from failing due to configuration issues in later stages.

Despite the benefits of this approach, there were two key limitations to its implementation. The first limitation was that there was a performance cost to generating new events in each stage. Specifically, new events required storage space within the SIEM event database. At a minimum, every interesting event or log that met stage one criteria would result in at least one more duplicate event being generated, all be it with a different classification and event title. In the worst-case scenario, a single event or log could meet the criteria for multiple stage one rules, and therefore result in a multiplicative increase in events. This was especially problematic in statistical baseline rules that were sensitive to drastic increases in events. Theoretically, one highly suspicious event could result in a storm of suspicious residual events. This phenomenon could be beneficial in prioritizing suspicious traffic but could also result in false positives and data retention issues if not configured properly.

The second limitation was that this type of rule was limited to classification labels available to SIEM engineers during rule construction. The LogRhythm SIEM classification labels, previously alluded to in Fig. 6, were not always easy to map to observed behavior. Technically, multistage rules did not require engineers to use the classification meta data field as primary criteria; however, doing so was helpful in ensuring at least one data source contained the necessary meta data fields for optimizing "group by" field inclusion when generating alerts. SIEM rule names could also be used as primary criteria, however this approach often resulted in commission of unexpected data sources which could have provided useful context when generating alarms. Labeling suspicious logs and events with similar classification labels allowed for data provided from multiple disparate data feeds to be used as criteria in multistage rules. If analysts requested that an alert contain certain pieces of meta data to provide context during triage, each of those fields could be used as "group by" criteria in establishing an event for that stage of an attack and remove logs or events that did not contain them. However, unlike the multi block rules, which exhibited a high false negative rate due to null queries for meta data within one or more rule blocks in series, if a rule block failed in a multi stage approach, that stage was simply omitted, but a terminal alert rule could still trigger off of other rules in different stages. Essentially, multistage rules could tag data within each stage, extract meta data if appropriate for said stage, and then be aggregated in an alert at the final stage of construction in necessary. Additionally, each stage could be used to satisfy multiple terminal rules, essentially creating several options for partial detection.

Although the multistage rule approach appeared to be the best option, it was seldom used by SIEM engineers. Creating these rules required a large amount of planning and a framework for guiding rule stages as well as classification labels. Unfortunately, the default SIEM classification labels were not appropriate for mapping to logical stages, and a consistent framework did not exist for describing attacker actions.

Table 1
SIEM alert names observed during blind penetration test.

Alarm Name	Count	Percentage
Critical Condition	436	48.77%
High Severity IDS/IPS Alerts	88	9.84%
Silent Log Source Resumed	68	7.61%
Password Modified By Another User	56	6.26%
Operations : Abnormal Log Volume Fluctuation Decrease	54	6.04%
LogRhythm Silent Log Source Error	53	5.93%
Operations : Abnormal Log Volume Fluctuation Increase	36	4.03%
Behavioral Anomaly : Host : Abnormal Authentication	23	2.57%
Account Disabled/Locked AIE Rule	20	2.24%
Critical Service Did Not Restart	18	2.01%
Successive Attacks	15	1.68%
Internal Brute Force from a Single Origin Host	7	0.78%
Internal : Suspicious : Multiple Accounts Disabled By Administrator	5	0.56%
Excessive Suspicious Activity	5	0.56%
External : Host Compromised : Attack/Compromise Followed By Process Starting	5	0.56%
LogRhythm Agent Heartbeat Missed	3	0.34%
Internal : Suspicious : Password Changed On Multiple Accounts By Administrator	2	0.22%

3.4. Significant data points

At least four data breaches, and more than a dozen blind penetration tests were observed during the study period. Each of the data breaches was initially undetected by security analysts and was escalated to the SOC by the client. Approximately half of the blind penetration tests were detected during the study period. All of the security breaches and penetration tests that did not result in client notifications resulted in demands for investigations from the client. These investigations required tier 3 analyst review for a period ranging from 48 hours to a week in order to determine root cause analysis and incident response actions.

3.5. Analysis of data collected

A few common themes were observed during these investigations. Every investigation identified some form of inadequacy in data provided to SIEM systems. In some cases, the compromised system was simply not sending data to the SIEM. In other cases, the system was sending data to the SIEM, but was not properly configured to audit actions in the level of detail required to detect the action (such as failing to audit process creation, or system log on events). In two instances the initial vector of compromise was social engineering resulting in the successful phishing of an employee's credentials and produced no observable data prior to the compromise. Several instances were identified where log data existed, but SIEM correlation rules did not fire based on strict threshold or sequencing requirements for alert triggering. In the final instance, a logically sound SIEM correlation rule could fail due to the lack of one or more data elements expected to be present in log data but was missing.

Aside from issues associated with data being collected by the SIEM, alarms configured to notify analyst were woefully vague. Table 1 depicts the unique SIEM alert titles presented to analysts during a blind penetration test. Over 48% of the alarms generated merely bore the title "critical condition". A few other alerts provided slightly better descriptions such as "high severity IDS/IPS alert", "successive attacks" and "external: host compromised: attack/compromise followed by process starting." As stated previously, none of these alerts were escalated to the client during the penetration test as they were deemed too generic to be actionable by SOC analysts. Analysts were able to investigate alarms and retrieve logs responsible for triggering the notification. In retrospect, the alarm "External: host compromised: attack/compromise followed by process starting" should have been investigated further and was likely omitted due to analysts being overwhelmed with other alarms during the observed period.

Analysts were routinely bombarded with an extreme volume of logs and alerts generated daily. 48 h worth of log data were retrieved from the SIEM during postmortem analysis of the penetration test discussed previously. The client in question had generated over 5.5 million logs during the collection period, resulting in over 700 alarms being sent to analysts for triage. None of the alerts generated during the blind penetration test were associated with "audit" classified data. Fig. 7 depicts the classification of logs contained within alerts generated during the blind penetration test. Note, none of the alerts generated contained logs classified as "audit".

Despite not being associated with any of the alarms generated for analyst triage, logs classified as "audit' by the SIEM accounted for over 55% of log data generated during the investigation period. Additionally, none of the alarms generated accurately reflected the actions performed by the security assessment team conducting the penetration test. Additional investigation into the nature of audit logs revealed that some of the most forensically useful information was contained within them, specifically the rarest occurring audit logs. Fig. 8 depicts the Windows audit logs that occurred most frequently during the penetration test, and Fig. 9 depicts the Windows audit logs that occurred the least frequently.

Local firewall data, specifically acknowledgment of allowed connections, accounted for over 1.7 million logs collected or approximately 55% of audit data. Account logon and logoff data accounted for roughly 45% of the remaining audit data. However, analysis of the least frequent audit events, accounting for 121 logs out of over 3 million collected during the investigation, provided detailed insight into actions performed by the security assessment team. Note, not all of these logs were associated with the assessment team; however, the actions being audited were much better suited for describing threat actions than the generic "security", "operations" or "audit" labels used previously.

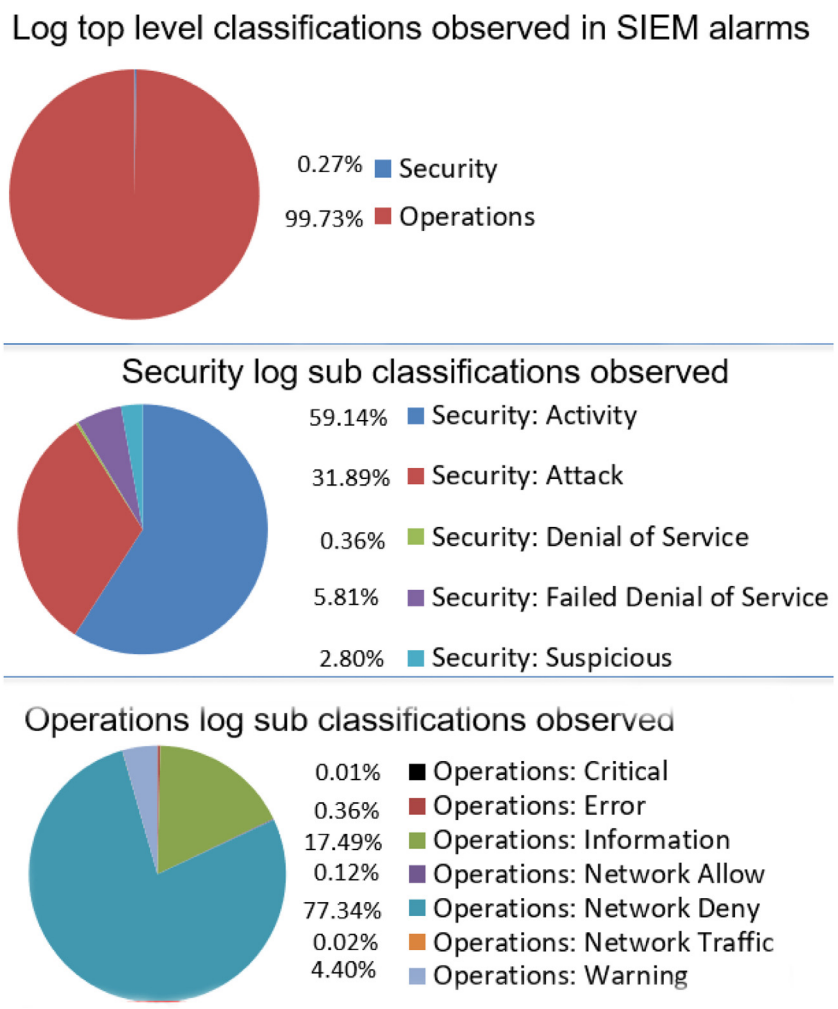

Fig. 7. SIEM event classification labels observed during blind penetration test (out of 2,573,479 logs observed).

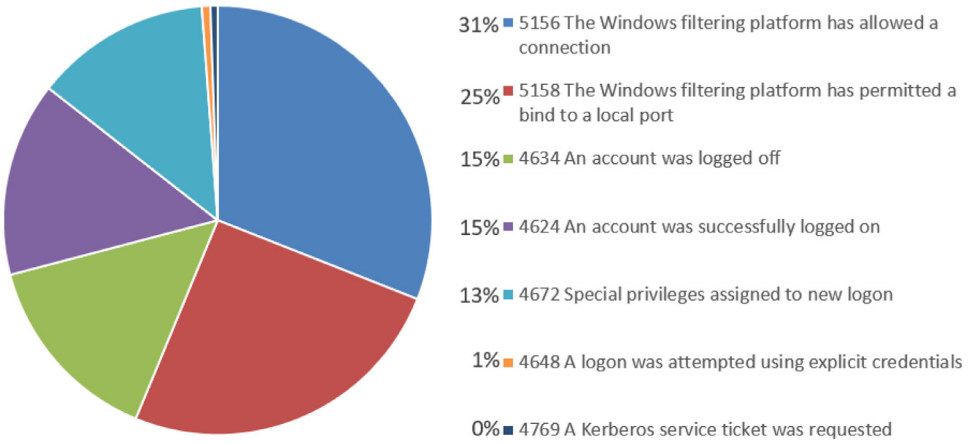

Fig. 8. Occurrence of most frequent Windows event IDs within audit data during penetration test (out of 3,146,389 Windows logs observed during investigation period).

4. Development of a novel SIEM configuration: Revising the log classification ontology

4.1. Overview

Observations during the SOC study indicated that merely generating SIEM alerts for suspicious activity did not increase the rate at which security analysts responded to events. In fact, it could be argued that too many SIEM alerts had an adverse effect on analyst response rates. Excessive alerts were either ignored by analysts or consumed vast resources to gather enough data to explain the nature of the alarms.

As was introduced within the SIEM engineer overview section of the SOC study, multistage rules could be used as a mechanism for increasing the amount of meta data contained within alerts as well as triage data sources during evolution through stages of

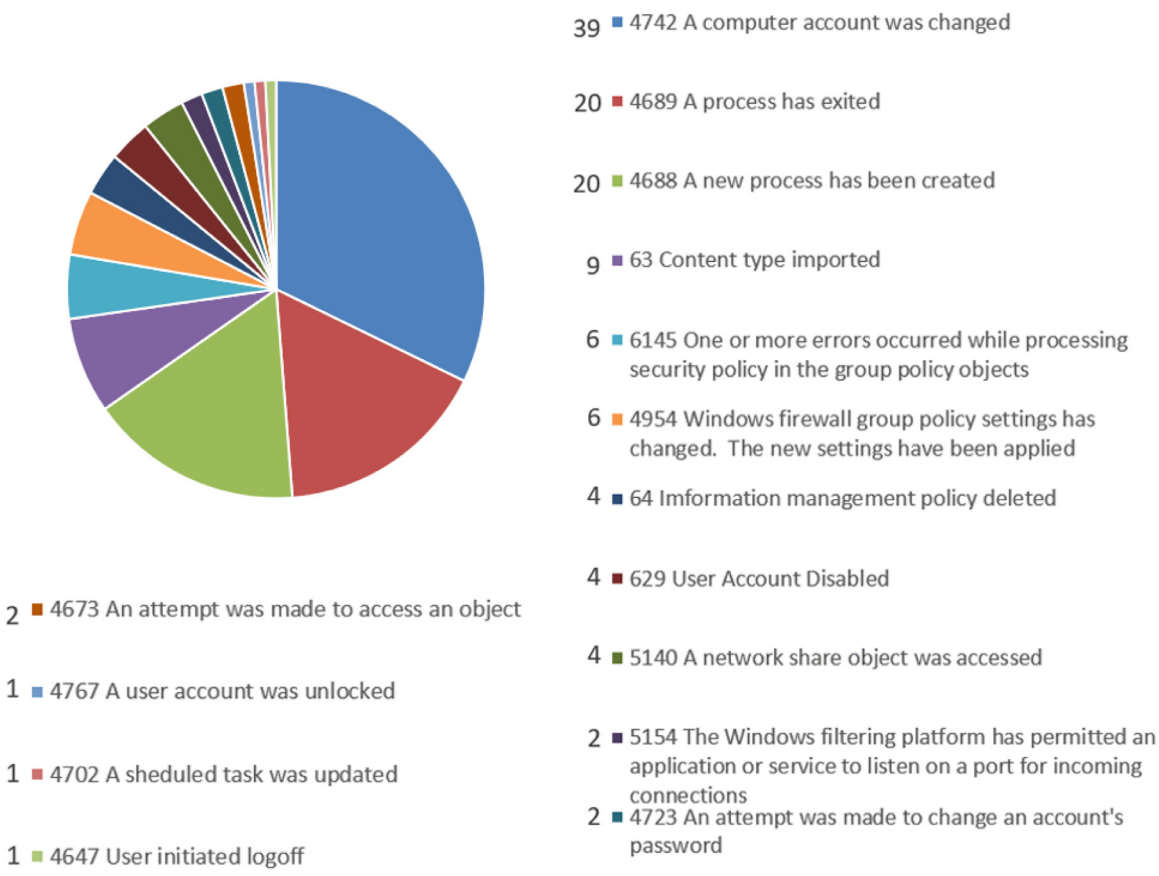

Fig. 9. Occurrence of rarest Windows event IDs within audit data during penetration test (out of 3,146,389 Windows logs observed during investigation period).

suspicion. However, data derived from confirmed breaches and penetration tests indicated that classification labels used in log parsing and event generation were inadequate for properly representing attacker actions on an ontological level. Furthermore, a standardized framework which could be used to describe attack stages had not been adopted between SIEM engineers, security analysts and clients.

The SOC study identified merit in being able to use broad classification labels when developing SIEM correlation rules from a SIEM engineering complexity viewpoint and alarm naming viewpoint for analyst triage. Ideally these correlation rule classification labels would align with kill-chain phases, and furthermore contain consistent metadata fields for reliable correlation with similar events in the same phase or allow for pivoting to data in heterogeneous phases. Unfortunately, none of the existing models, discussed in the background section, were ideal for suiting both human investigative processes and automated SIEM correlation.

The Lockheed Martin, Mandiant and Mitre models each developed a framework conducive to categorizing observed data through the perspective an adversary, which seemed to mirror the logical investigative process SOC analysts migrated toward during incidents. However, none of these models considered sequencing or metadata requirements for automated SIEM correlation rule development. The Mitre framework provided additional detail by including adversary techniques and potential indicators of compromise associated with each objective phase, but Mitre phase categories do not neatly align with consistent sensor metadata groupings, and therefore could have been problematic in use as classification labels in multi stage rule construction.

Despite the lack of a “drop in” solution for a correlation framework, the kill chain approach appeared to be consistent with analyst observations that recurring patterns of evidence emerged within certain phases of an attack lifecycle. A framework representing different attacker objectives, tasks, and related forensic data was created to serve as a new SIEM log ontology based upon these observations, deviating from either of the “kill-chain” models described previously Bryant and Saiedian (2017).

4.2. Development of the novel framework

The proposed framework was initially inspired by the Lockheed Martin kill chain and the Mandiant APT1 attack lifecycle models that were published circa 2013. These frameworks were used as the basis for investigating routine escalations or performing post-mortem analysis of key events during the SOC study. Note, these actions were primarily associated with tier 2 and tier 3 analyst activities and not routine tier 1 analyst triage. Additionally, many of the phases depicted by these frameworks were not compatible with creating automated alert logic via SIEM systems, ultimately requiring manual review of raw log data.

After several months of investigations while referencing the Lockheed Martin and Mandiant frameworks, it became apparent that neither model was suitable for creating strict deterministic correlation rules similar to the ones used in the default SIEM rule base. For instance, no clean rules existed for determining that an “initial compromise” had occurred. It was understood that a compromise must take place to access the system, but opinions among analysts varied as to what indicators would reliably indicate a system was compromised without having a large number of false positive cases. In such instances where a compromise was suspected, or alleged by a client demanding an investigation,

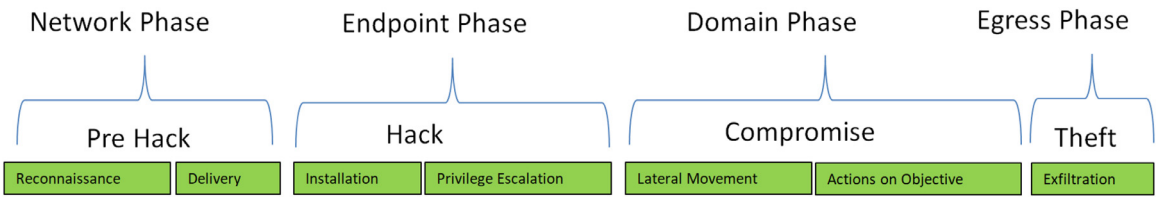

Fig. 10. Investigation framework phases derived from the four logical domains of: network, endpoint, domain and egress.

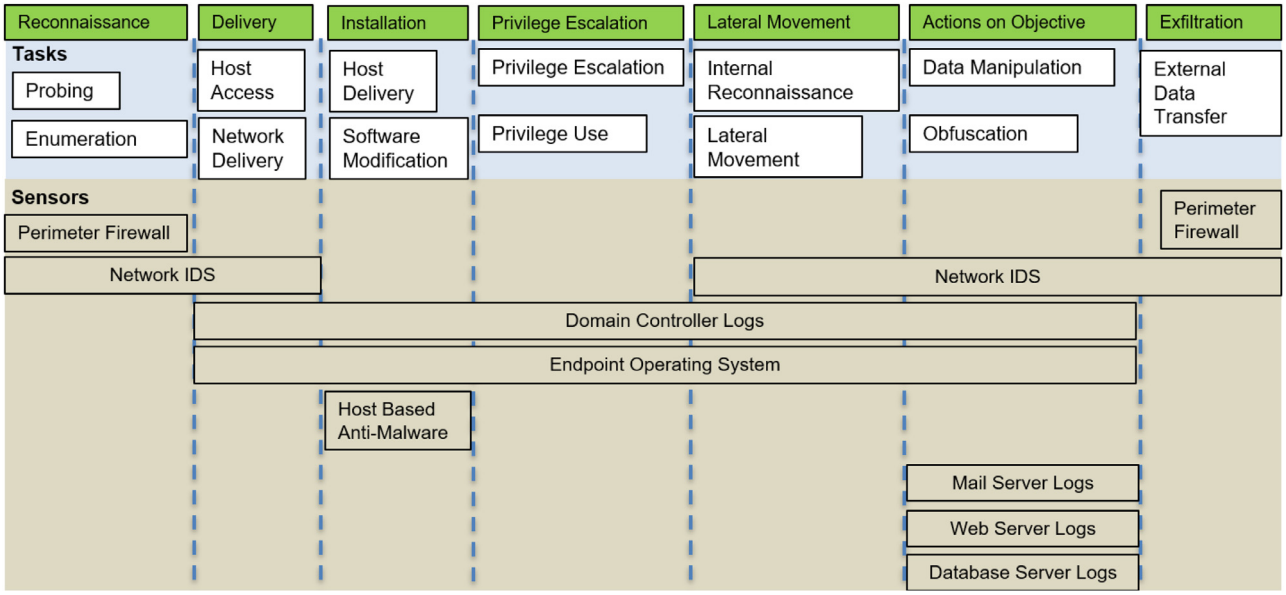

Fig. 11. Data source alignment with novel killchain phases.

analysts immediately began collecting information on the alleged compromised system and pivoting off of metadata identifiers such as IP addresses, host names, account credentials, port numbers, or processes to confirm suspicious activity was occurring around the system in question.

During this investigative process, distinct phases were identified with homogeneous metadata provided reliably by sensor feeds. Additionally, these phases could be aligned with discrete and descriptive actions, rather than generic labels such as “compromise” or “exploit.” Aligning reliable metadata sources with logical pivot phases was imperative for analyst data queries in attempts to make root cause determination for a suspected breach and could be used to perform automated pivots and data aggregation by the SIEM via multistage correlation rules.

The novel model was eventually developed around seven descriptive phases: reconnaissance, delivery, installation, privilege escalation, lateral movement, actions on the objective and exfiltration depicted in Fig. 10. These phases were further grouped into four logical domains: network, endpoint, domain and egress, based on the types of systems that provided reliable metadata for detection and correlation. Fig. 11 depicts phase alignment with reliable data sources for each phase. The egress phase included both network data and endpoint data, but was differentiated by specific adversary tactics or techniques

Many of these phases align with phases of the Mitre ATT&CK™ framework, with the exception that the Mitre framework appears to focus on end point analysis and omits phases that cleanly align to network device data, such as “reconnaissance” and “delivery”. It is worth noting that network device data could prove useful to pivot from data associated with a compromised end point observed network traffic useful for determining the initial entry into the network, or other potential pivot points.

5. Applying the kill chain to SIEM software

5.1. SIEM platform selection

The LogRhythm commercial SIEM platform was selected as the preferred system to evaluate inclusion of a kill-chain model based on the authors’ 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 since none of them operated in a multi-tenant fashion and fewer SOC analysts were dedicated to triaging alerts generated within these systems.

The LogRhythm system consists of multiple distinct data processing subsystems. The first subsystem, referred to as the log manager, is responsible for initial data ingestion and parsing free text into normalized metadata fields. The second subsystem, the event manager, is responsible for creating and labeling interesting “events” based on normalized log data or other events generated by the SIEM. The third subsystem, the advanced intelligence engine (AIE), is responsible for implementing advanced correlation logic to logs or events previously generated by the SIEM and generating new events with custom names or labels. Either the event manager or AIE subsystems may be used to generate alerts by the system; however, the AIE subsystem is the only subsystem capable of implementing multiple logic blocks and was therefore the preferred subsystem for developing correlation rules within LogRhythm.

The LogRhythm data flow 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 during the initial

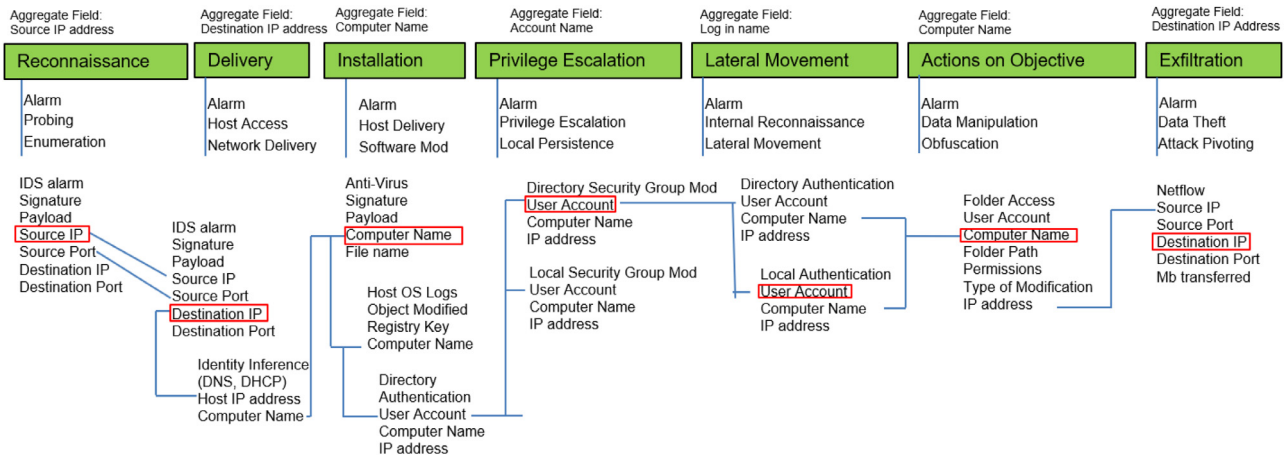

Fig. 12. Reliable Pivot Metadata by Killchain Phase.

parsing and normalization phase by the message processing engine within the log manager, or by the AIE during correlation and subsequent reclassification. Fig. 6 illustrates the default classification labels available for classifying log and event data.

The ability to use the AIE to mutate a log or event into a new event with a custom classification label was a critical component in implementing the novel framework within the SIEM. This provided a mechanism for creating a new event comprised of metadata contributed by dissimilar but related sources and attributing this newly created event to specific phase-aligned activity. This discovery became a fundamental element in implementing data enrichment and aggregation within the SIEM platform. Fig. 13 depicts the introduction of new classification labels within the LogRhythm SIEM conducive to alert creation and log aggregation within subplanes and associated alert phases.

Unfortunately, the AIE subsystem relied upon specific SQL queries to populate metadata fields in events or alerts it created. For instance, if one desired to include the account name, process name, source IP address and destination IP address associated with a remote access program, each of these fields must be included as "group by" fields within the AIE SQL query. If any of these desired metadata fields were null, then the entire query would fail, resulting in a false negative. It was very rare for all desired metadata fields to exist within a single data point sent to the SIEM system. Furthermore, since each desired metadata field was required by the AIE system, data points with partial detections would be omitted from an AIE event or alert. Partial detections assisted in providing context to analysts and providing insight to root cause attribution. Therefore, it was necessary to aggregate metadata from multiple partial detections to be included within AIE events or alarms.

Aggregation with the new model was performed in a multi-staged process described in the following sections. The first aggregation stage was associated with combining related metadata elements from dissimilar sources to create intermediary events suitable for AIE correlation. The second aggregation stage was associated with combining events and alerts into a manageable number of notifications to be sent to human analysts for review.

5.2. Data enrichment and intermediary event construction

Analyst postmortem analysis of known security events indicated a need to use specific metadata fields within each phase to pivot between log sources and generate an accurate depiction of suspected malicious actions. This observation shaped the development of the new hybrid model wherein each newly devised phase has a natural "aggregation" metadata field depicted in Fig. 12.

For instance, "reconnaissance" aligns naturally to network data of the source machine (i.e. source IP address), while "delivery" aligns naturally to network data of the destination machine (i.e. destination IP address). These fields (source IP and destination IP) were normalized by the SIEM log manager sub-system during log ingestion. Later stages, such as "installation", could potentially contain several types of metadata, from network data (IP and MAC address) to user information (account names, privilege levels, security groups). These stages were therefore segregated based on which metadata fields were most pertinent to detecting and describing action within them, rather than where the detection occurred.

It is possible for a single data source or event to contribute to multiple phases, such as "installation" and "privilege escalation" events. Both phases are likely to be observed within endpoint operating system logs. However, knowing that "installation" logs are machine based (e.g. aggregation is conducted on hostname or other computer identifier) and "privilege escalation" events are account based (e.g. aggregation is conducted on account/username), provides insight as to how to best combine data within their respective logical phases.

Ideally distinct metadata from multiple dissimilar data sources could be combined automatically by the SIEM using these natural metadata pivot points. The hybrid model was originally designed so that each phase would contain metadata fields necessary to perform automated event combination. Fig. 12 depicts the natural metadata fields conducive to aggregating logs or events from disparate sources within the same logical phase. This relational database approach was motivated by the fact that the LogRhythm SIEM utilized SQL queries to perform correlation functions. However, phase labeling may prove beneficial to other systems that are not using the SQL language.

Unfortunately, not every data source provided all metadata fields necessary for proper correlation via the AIE. It became apparent that it was necessary to create intermediary events that combined disparate metadata fields from dissimilar but related sources within phases. This observation led to the development of sub-phases within the hybrid model. Each parent phase of the hybrid model was expanded to include three sub-phases. Two of the sub-phases were designed to label log data that contained partial elements of ideal metadata fields and a third sub-phase was designed to indicate primitive alerts within the parent phase.

Partial data sub-phases provided an elegant solution for suspicion escalation in a manner similar to the subplan-based correlation scheme described by Chien et al. (2007). These events were not suitable for creating alarms based solely upon the data they

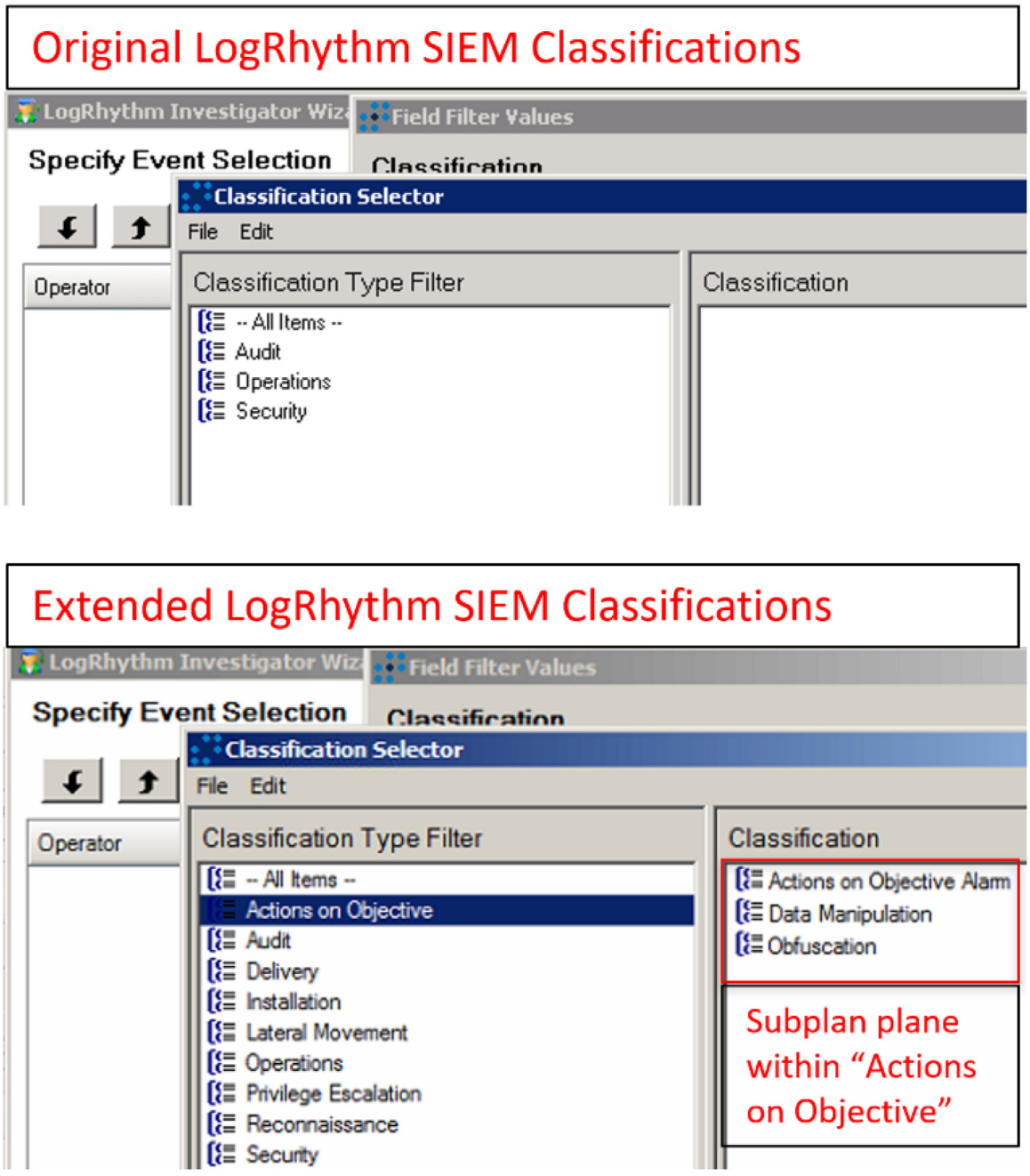

Fig. 13. Comparison of SIEM classification labels after modifying the LogRhythm database .

contained; however, they were beneficial for combination with other related events to create a more complete picture of activity within the network. Events tagged with a partial data sub-phase could easily be combined with other events with the same, or adjacent sub-phases to create more complete events containing ideal metadata fields. The reason two sub-phases were used for this purpose was to allow for data sources to potentially provide insight to transition between parent phases. Knowing that events with specific classifications are guaranteed to contain certain metadata fields provides for the development of reliable correlation rules and transition between logical phases of the model.

An example of adjacent phase transition with sub-phase labeling may include network-based IDS data which contributes to two phases of the hybrid model, "reconnaissance" and "delivery." The aggregate metadata field associated with the "reconnaissance" phase is the "source IP" field, while the aggregate metadata field for the "delivery" parent phase is the "destination IP" field. As the sub-phase of "enumeration" exists on the border between "reconnaissance" and "delivery", it must contain both "source IP" and "destination IP" metadata fields at a minimum. Further extending the network-based IDS example, IDS signatures attributed to interrogating services running on target systems could be combined with data attributed to the "host access" sub-phase of "delivery." "Host access" classified data must contain "source IP" data as well as "destination IP" data. Therefore, response or firewall allow messages from a victim machine should be classified as "host access" to allow for natural correlation between "reconnaissance' and "delivery" phases of the hybrid model. Fig. 11 shows how data sources may contribute to phase transitions within the hybrid model.

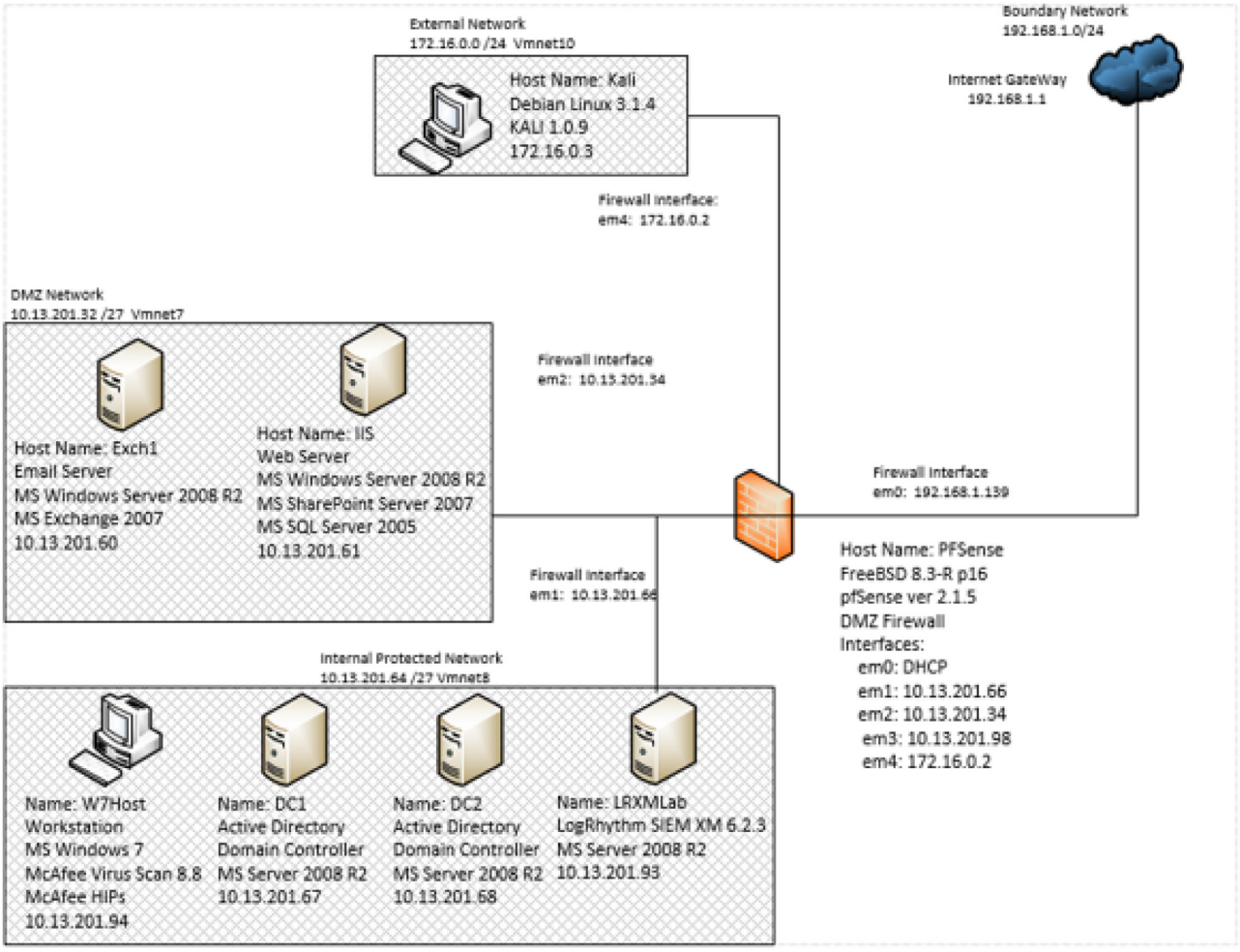

Fig. 14. Laboratory logical architecture depicting network configurations and IP space.

It is possible that a single event may contain all ideal metadata associated with a parent phase, meaning it satisfied minimal metadata fields for both child sub-phases. In such cases, the event would be classified with the parent phase classification, such as "privilege escalation." Events labeled with the parent phase classification are referred to as intermediary events. Even though intermediary events contain the ideal amount of metadata fields, they may not necessitate generation of an alarm or analyst notification. Intermediate events were primarily intended to enrich other alarm generating events with as many metadata fields as possible to contextualize the alarm. Additionally, intermediary events were ideal for combination with other intermediary events in adjacent phases, as they would contain minimal metadata fields for combination with both the phase preceding and following the phase in which they were associated with.

The "alarm" sub-phase label was reserved for events that were well known or high confidence indicators of phase activity. These events were worthy of generating an alert in and of themselves but may benefit from additional context through combination with partial data sub-phases or intermediary events. Once an event was generated with the "alarm" sub-phase label, all other partial or intermediate events within that phase would be combined into a single alarm leveraging the natural aggregate field associated with their parent phase. The resultant alarm would be presented to an analyst with all distinct metadata values associated with phase classifications, ultimately automating several secondary or tertiary investigations analysts would have previously performed manually. A more thorough explanation of alarm construction for metadata and alarm aggregation is reserved for the following section.

5.3. Alert fusion

The LogRhythm SIEM allowed for two different approaches to creating alerts to be sent to analysts for triage. The "traditional" approach leveraged the event manager subsystem of the SIEM and was only capable of performing pattern matching queries for metadata fields contained within events stored within the event database. However, despite this limitation, the "traditional" approach performed a greedy query and all records returned would contain data in any metadata fields that were not blank. The second approach to creating alerts leveraged the AIE subsystem.

The AIE subsystem presented several advantages over the traditional approach, but also had several key limitations. Advantages of the AIE system over the traditional approach included the ability to conduct statistical baselining as well as the ability to create a series of conditions that must be met to trigger an alarm. Additionally, the AIE subsystem was able to extract data from either logs or events. Logs represented the purest form of data in the SIEM and consisted of a copy of the free text message sent to the SIEM as well as parsed and normalized metadata fields extracted from said message. A classification label was still assigned to log data, however generating an alert from log data was not possible in the traditional approach; only event data could be leveraged to generate traditional alarms. Events were often comprised of one or more

logs and contained an additional metadata tag used to describe the event called a "common event name." The intent of converting logs into events was to allow for innocuous or chatty logs to be immediately archived and decrease the amount of data the alarm engine would need to travers to generate alerts. The AIE engine could convert one or more logs into intermediary events, as was described in the previous section of this paper.

The primary limitation of the AIE subsystem was that it did not perform greedy queries and would only produce alerts or events if all metadata fields contained within conditional statements were present within log or event data. This meant that the AIE subsystem was prone to a high false positive rate if intermediary events were not created prior to attempting to implement more advanced correlation rules or conditional statements in series. Additionally, if a query succeeded, only the fields requested would be returned. If a log or event that satisfied alert criteria contained additional metadata fields, not contained within the AIE conditional statements, this data would be omitted from the final alert resulting in the need for analysts to perform additional manual queries to retrieve context for the alarm.

Despite its limitations in capabilities for advanced correlation logic, the "traditional" approach was ideal for aggregating data from multiple events or alarms into a single consolidated notification sent to analysts. This complemented the AIE subsystem well in that it provided a mechanism for implementing greedy data retrieval of related events, while the AIE subsystem provided the ability to construct intermediary events and apply custom classification labels. Alerts generated using the traditional approach contained a summary of all distinct values for every metadata field and could be grouped by one or more "aggregate" fields that needed to be identical across records.

The most simplistic implementation of event fusion was to create a "traditional" alarm that queried for the presence of any event with the "alarm" sub-phase classification, then combine related events based on identical values in the natural aggregate field for the phase the "alarm" label belonged to. The default labels depicted in Fig. 6 unfortunately lacked such alarm labels, or the level of detail necessary to implement the metadata aggregation scheme described in the previous section of this paper.

Ultimately, the two-staged process of leveraging the AIE subsystem to perform intermediary event construction and using the traditional alarm approach to aggregate metadata fields was adopted as the preferred method to alert construction. The use of novel classification labels associated with the hybrid model was essential to proper event construction and aggregation when using this approach.

6. Testing and evaluation of the novel SIEM configuration

Other works have established evaluation frameworks to comare disparate SIEM systems Safarzadeh et al. (2019). However, the data environment available to this study predominantly relied upon a specific SIEM plaftorm, namely LogRhythm. As such, the evaluation and testing performed in this work was limited to implementing the novel framework in just one platform and did not implement comparative frameworks such as the one devised by Safarzadeh et al. (2019)

A sophisticated network security laboratory environment was designed to evaluate the efficacy of the SIEM configuration modified with a novel ontology and is depicted in Fig. 14. Two identical laboratory environments were constructed with the single variable between deployments being modifications to the SIEM database used to detect security events. This section focuses on the design of the laboratories and the details of the experiment for which they were used.

6.1. Laboratory network design

A virtual network was constructed to evaluate baseline and enhanced SIEM configurations. Two separate but identical virtual environments were constructed, with the exception that the LogRhythm SIEM system in one environment was configured with vendor recommended default correlation rules and the other environment contained a LogRhythm SIEM system enhanced with additional classification fields reflecting the hybrid kill-chain model.

Microsoft operating systems were selected as the basis for the majority of virtual systems within the laboratory network due to security analysts' familiarity in conducting forensics investigations based on Microsoft technology as well as a more robust library of default SIEM correlation rules designed for Microsoft systems. Services hosted on Microsoft systems included: directory services, email hosting, web services, and a SQL database. A suite of McAfee anti-malware products was deployed to endpoints to provide antivirus and host based intrusion prevention system data via a centrally managed server. A pfSense virtual machine was deployed to serve as a virtual layer three device, necessary for network traffic shaping, as well as a platform to host open source security tools including: Snort IDS, Squid proxy, and network based firewall capabilities. All Microsoft endpoints were configured with host-based firewall settings to provide an additional layer of security beyond network based filtering as well as provide supplementary data for correlation with data provided by network centric sensors. Audit policy settings on all Microsoft endpoints were adjusted to provide additional forensic details omitted by default configuration settings, such as logging network traffic denied by host-based firewalls or process creation.

6.2. Attack experiment design

Real world security breaches do not always reflect every stage represented by the hybrid kill-chain model. As such, a custom scenario was devised to stimulate sensors and ensure coverage of all seven stages. This scenario combined traditional reconnaissance and probing techniques, indicative of opportunistic attacks, as well as targeted attacks typical of advanced persistent threats. Table 1 depicts the types of actions that were performed during the attack scenario.

6.3. Detection rate comparison

The modified SIEM ontology outperformed the baseline SIEM ontology in alert metrics with a 96% true positive detection rate by generating an alert for 25 out of 26 test scenarios. The baseline SIEM ontology and LogRhythm default rule set had a 26.9% detection rate with alerts generated for 7 out of 26 of the test cases. Additionally, the modified ontology generated aggregate alerts with metadata from multiple events for 76% of alerts (19 of 25). The remaining six alerts were associated with singular events where no additional data was available for aggregation.

It is worth noting the difference in alert volume in addition to improvements in the true positive rate. The baseline SIEM generated 83 alerts during the evaluation, however they were only associated with 7 of 26 test cases. An open vulnerability assessment system (OpenVAS) vulnerability scanner test case resulted in nearly half of the baseline SIEM alerts with 41 separate alerts. Conversely, the modified SIEM generated five alerts during the same test case, containing aggregate metadata from 401 correlated events, and 46 alerts from all test cases. This data indicates the ability to aggregate data via a logical identifier metadata field proved to be an effective mechanism for decreasing alert volume. A detailed comparison of alerts between the baseline and modified SIEMs are listed in Table 2.

Table 2
Comparison of baseline and modified SIEM alert performance: Alert comparison.

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

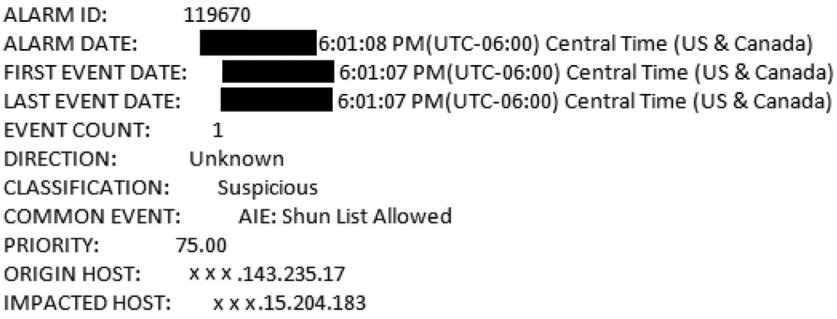
ALARM ID: 119670
ALARM DATE: [redacted] 6:01:08 PM(UTC-06:00) Central Time (US & Canada)
FIRST EVENT DATE: [redacted] 6:01:07 PM(UTC-06:00) Central Time (US & Canada)
LAST EVENT DATE: [redacted] 6:01:07 PM(UTC-06:00) Central Time (US & Canada)
EVENT COUNT: 1
DIRECTION: Unknown
CLASSIFICATION: Suspicious
COMMON EVENT: AIE: Shun List Allowed
PRIORITY: 75.00
ORIGIN HOST: x x x .143.235.17
IMPACTED HOST: x x x.15.204.183

Fig. 15. Typical email alert generated by baseline SIEM.

6.4. Alert forensic value comparison

The primary motivation for developing the new SIEM ontology was to provide a mechanism for the aggregation of pertinent and related metadata into alert notifications to decrease the investigative effort associated with explaining security alerts. The baseline SIEM ontology combined 47 OpenVAS test case alerts into a single email containing 7154 words. It was not obvious which metadata field was used to correlate these events, since none of the fields were common across all 47 alerts. The email batching process merely listed alerts, rather than combining them in a logical manner. Only 41 alerts were generated within the analyst GUI console during the OpenVAS scan test case, indicating six additional alerts must have been aggregated from previous scan activity. It appears this aggregation was most likely performed based on the large increase in alerts generated within a short time frame during the scan, resulting in combination based on temporal proximity, rather than through metadata correlation. Many of the alerts

LogRhythm Alarm - Lateral Movement Events Observed by Account

LogRhythm@siem.lab.com
Sent: [redacted]
To: user

```
ALARM ID:                141064
ALARM DATE:              [redacted] 8:23:32 PM(UTC-06:00) Central Time (US & Canada)
FIRST EVENT DATE:        [redacted] 8:23:11 PM(UTC-06:00) Central Time (US & Canada)
LAST EVENT DATE:         [redacted] 8:23:11 PM(UTC-06:00) Central Time (US & Canada)
EVENT COUNT:             6
DIRECTION:               Unknown
CLASSIFICATION:          Lateral Movement Alarm
COMMON EVENT:            AIE: ALRM_Internal_Reconnaissance_Tools_Observed, AIE:
EOI_Windows_RDP_Logon_Successful
PRIORITY:                61.00
ORIGIN HOST:             W7HOST * (10.13.201.94)
IMPACTED HOST:           W7HOST *, DC1 *
ORIGIN LOGIN:            administrator
ALARM RULE NAME:         Lateral Movement Events Observed by Account
ALARM RULE DESCRIPTION:
MPE RULE:                C EVID 4688 & 4689 : Process Startup And Shutdown, C EVID
4624 : Administrator Logon Type 10
ENTITY (ORIGIN):         Global Entity, Internal_LAN
ENTITY (IMPACTED):       Internal_LAN
ACCOUNT:                 administrator
LOG SOURCE INFORMATION:  AI Engine Server (LogRhythm AI Engine)
ZONE (ORIGIN):           Unknown, Internal
ZONE (IMPACTED):         Internal
TCP/UDP PORT (ORIGIN):   3315\?
ALARM RULE ID:           1000000140
VENDOR MESSAGE ID:       4688, 4624
PROCESS:                 mstsc.exe, nmap.exe, ping.exe
URL:                     c:\windows\system32\, c:\program files (x86)\nmap\

LOG MESSAGES
------------------------------

<aie v="1"><_0 Account="administrator" CommonEventID="10404" DHost="1|3||6|w7host"
MPERuleID="1000000534" MsgClassID="1400" MsgSourceHostID="6" NormalMsgDate="[redacted]
[redacted] NormalMsgDateLower="

<aie v="1"><_0 Account="administrator" CommonEventID="10404" DHost="1|3||6|w7host"
```

Fig. 16. Email alert generated by modified siem depicting log aggregation and framework phase attribution.

in the batch of 47 alerts generated during the OpenVAS scan correctly identified abnormal net-work connections to the Windows 7 host W7host with IP address 10.13.201.94, as was replicated during the scan; however, no additional information was provided to indicate which computer(s) were attempting to communicate with the workstation, nor what aspect of the communication was considered abnormal. An analyst would be required to review all 47 alerts generated in order to identify the attacking machine or the scope of the probes conducted within the network.

Ten of the alerts in the pool of 47 correctly identified the attacker machine as the origin host with IP address 172.16.0.3, but it was not obvious what actions this host was conducting within this batch of alerts. One alert indicated a machine with IP address 172.16.0.3 was suspected of being associated with a system compromise or lateral movement, but there were no metadata artifacts associated with the alert to indicate how the conclusion was reached. In reality, the attacker had not yet successfully compromised a machine at this point.

Four of the 47 batched alerts indicated suspicion of a port scan, but only one of these four alerts indicated both the source and destination machines associated with the port scan activity. The remaining three alerts only indicated the targeted machine. Figs. 15–17 compare email alerts generated by the baseline SIEM configuration and the modified SIEM with extended classification labels.

Modified SIEM ontology email alert analysis In contrast to the 47 batched alerts generated by the base-line SIEM ontology, the modified SIEM ontology accurately identified the scan activity with a single alert. This was achieved by aggregating metadata fields from multiple events within the alert. The event field within the alert shows that 92 related events were combined. All alert notifications generated in the baseline configuration were comprised of a single event, even when batched. The modified SIEM alert title, depicted in the email subject line, identified the event as being associated with suspected reconnaissance activity and the aggregate field for correlation was the origin host field. The origin host, was correctly identified as the Kali Linux machine with IP address 172.16.0.3. The entire list of targeted machines was provided within the alert.

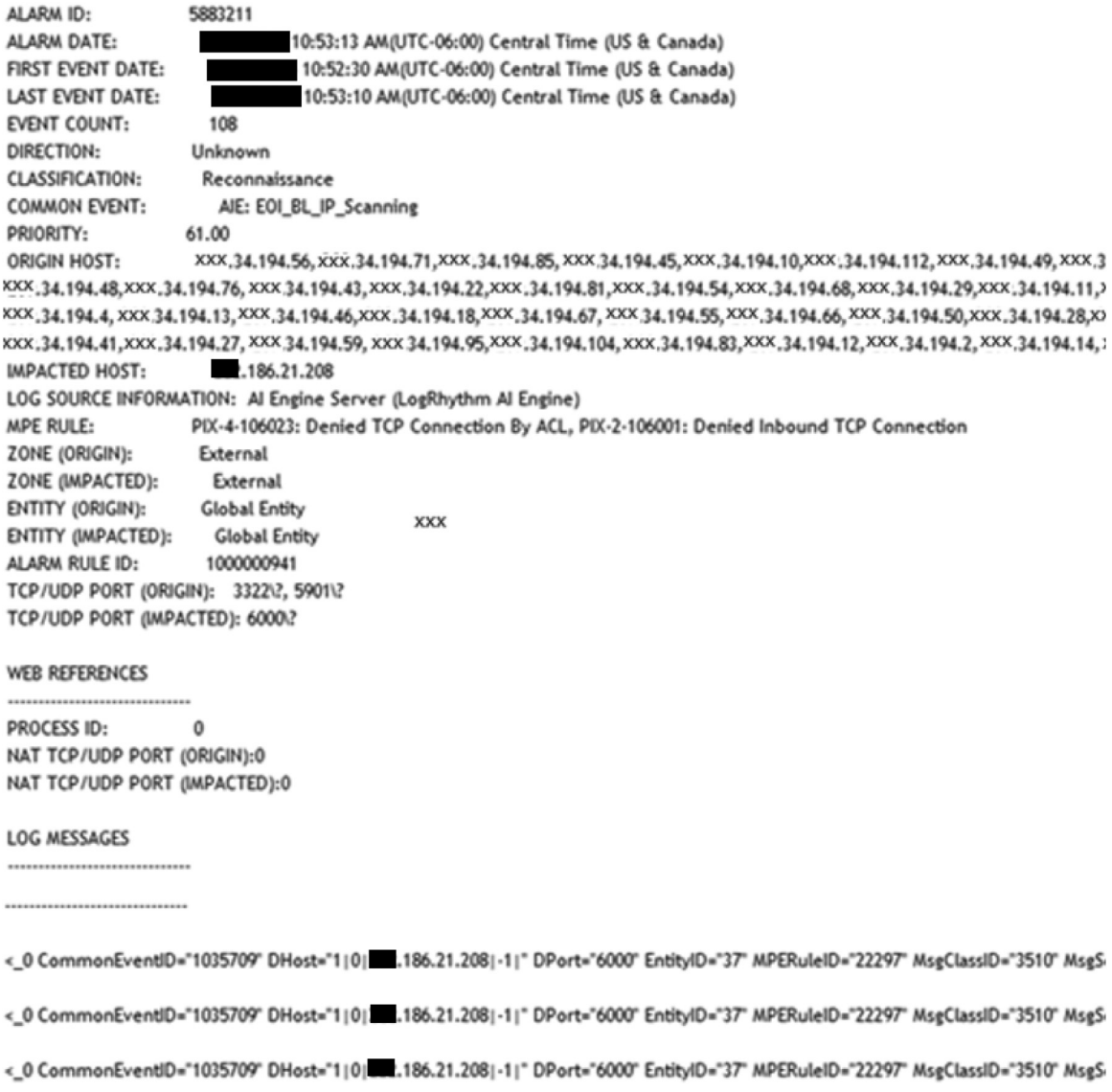

Fig. 17. Email alert generated by modified siem depicting log aggregation.

Supporting metadata, including port numbers and names for aggregated events, were also provided.

Alert forensic value conclusions

The modified SIEM alerts provided considerably more correlated data than the baseline SIEM alerts. As a result, security analysts were more likely to receive enough information to draw conclusions regarding the nature of the activity, requiring fewer manual queries to validate their hypothesis. The alerts presented using the baseline SIEM configuration often required a considerable amount of analysis of similar alerts to determine what data was actually detected and what data may warrant additional investigation. From a forensic perspective, the data contained in the modified alerts was superior to the data contained in the baseline SIEM alerts.

Email alert volume comparison

The baseline SIEM configuration generated 2364 alerts during experiment period, averaging 100 alerts per day. Conversely, the modified SIEM configuration generated eight alerts from the continued experiments, averaging one alert per day. The decreased alert volume may be attributed to the decreased number of detection rules configured between the two deployments. The modified SIEM had less than a third of the rules of the baseline SIEM, and 99% fewer alerts when test data was not being generated. The baseline SIEM rules generated an average of 0.78 alerts per rule per day, while the modified SIEM rules generated an average of 0.025 alerts per rule per day. In light of the modified rule set's improved true positive detection rate, it is determined the decrease per alert rule rate during non-testing conditions reflects a decreased false positive rate.

6.5. SIEM rule complexity comparison

The baseline SIEM rule set consisted of 128 correlation rules while the modified SIEM rule set consisted of 39 rules. This was achieved by segregating rules into separate groups consisting of specific event queries and aggregate alarm queries, while the baseline SIEM configuration used only specific queries. The decreased number of queries required to detect threat actions is assessed to be an improvement over the base model due to the assumption that fewer administrative actions will be required by SIEM engineers to maintain the system. Additionally, the queries contained within the modified SIEM rule set hierarchy were generally less complex than the baseline rule when compared side by side.

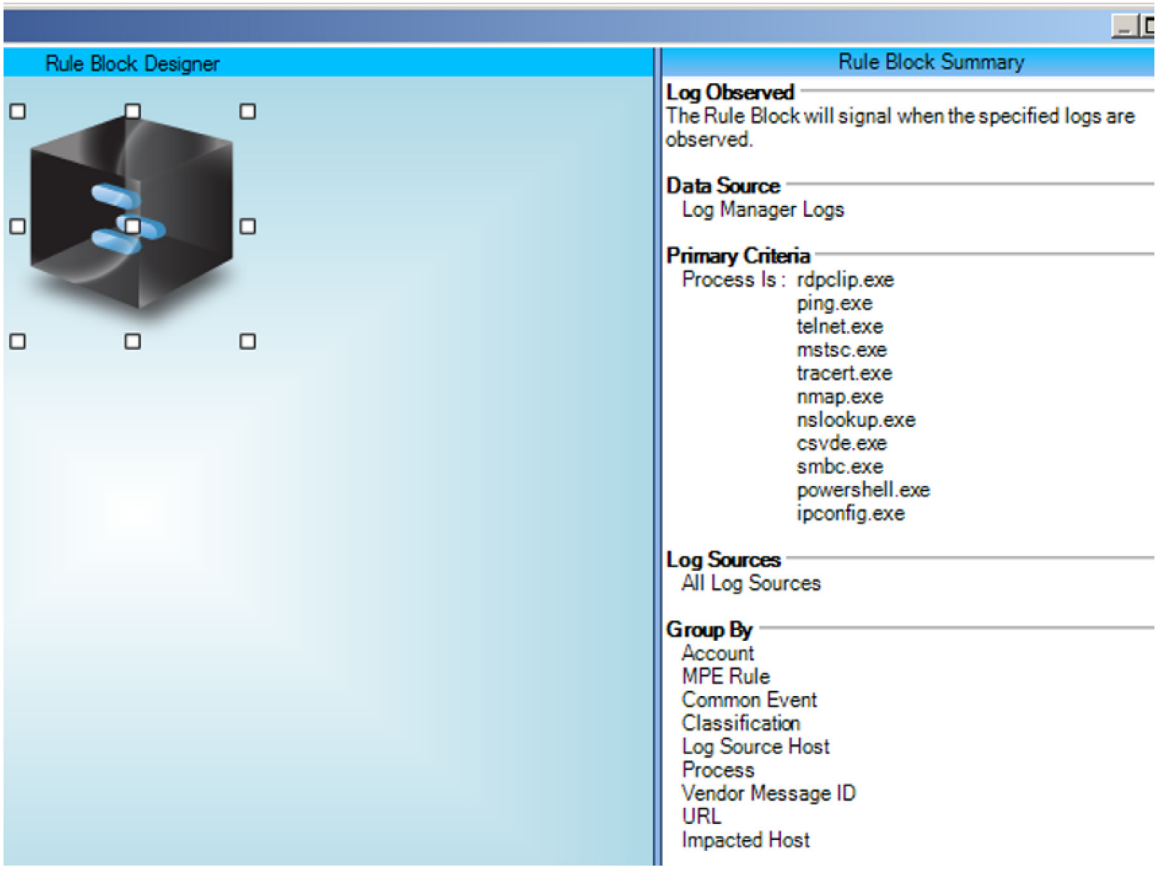

Fig. 18. Modified SIEM rule constructed using static indicators to generate an "internal reconnaissance classified event.

Baseline SIEM rule complexity analysis Many baseline SIEM rules leveraged statistical analysis rules implementing a baselining or learning logic block and a threshold comparison logic block. The baselining stage constructs a dynamic list of unique values during the learning period, which is configured to be seven days by default, and generates an average number of unique values observed by host. The threshold stage searches for deviations from the baseline. One such rule searched for more than five unique processes running in memory beyond the average determined by the baseline. This rule consumed approximately 17% of the memory allocated to the Advanced Intelligence Engine service running on the SIEM.

Some baseline SIEM rules were based off of the multi block approach, intended to observe a series of discrete events and generate alerts based on "if this then that" logic blocks. However, these types of rules suffered from several limitations. The first problem with these rules was their inability to tolerate the absence of an expected event in the rule chain. For instance, a rule designed to indicate system compromise by triggering an alarm if an administrator logon event with Windows event ID 4672 occurred following several failed logon attempts would fail to trigger if no administrator logon event with Windows event ID 4672 was logged by the target system. However, there may be several indicators of an administrator logging on to a system, which may need to be evaluated, especially if the target system was not configured to explicitly log privileged system accounts. A whitelist of known administrator accounts could be used as a reference for analysts to verify administrative access for instance. Using a whitelist/blacklist instead of the Windows event 4672 event would require the creation of an additional SIEM rule, as there is no elegant mechanism for informing the SIEM that a 4672 event and a whitelist/blacklist of administrator account names are equal events.

The second limitation of the traditional "if-this-then-that" logic approach with the basic SIEM structure is the requirement for strict sequencing of events. SIEM engineers are required to make several different correlation rules anticipating every possible combination of adversary actions in order to trigger complex rules. To further complicate matters, data from multiple systems may vary with incorrectly synchronized system times between data sources. This could create the illusion that events occurred out of order, and confound correlation rules based on strict sequencing.

Modified SIEM rule complexity analysis. Unlike the baseline SIEM rules, modified SIEM rules implemented a multistage approach and leveraged intermediary events to satisfy rule conditions. Event criteria in each stage mapped to static lists of indicators observed while using the novel framework during investigations, rather than a baselining mechanism. Using a static indicator list removed the need to implement a memory intensive baselining and threshold establishment. For example, a modified rule designed for detecting rogue processes was configured to generate an event within the SIEM event database for any process name observed but not listed on the static list of approved processes. The memory resources consumed by a modified SIEM rule query were negligible and reported as 0% of the total resources available to the Advanced Intelligence Engine process running on the SIEM. This is a marked improvement over the baseline SIEM rule constructed to perform the same function.

Additionally, since the modified SIEM was expanded to include several new descriptive classification labels, rules could be configured to categorize observed logs as either interesting events, or alerts requiring analyst triage. Unlike the baseline SIEM rules,

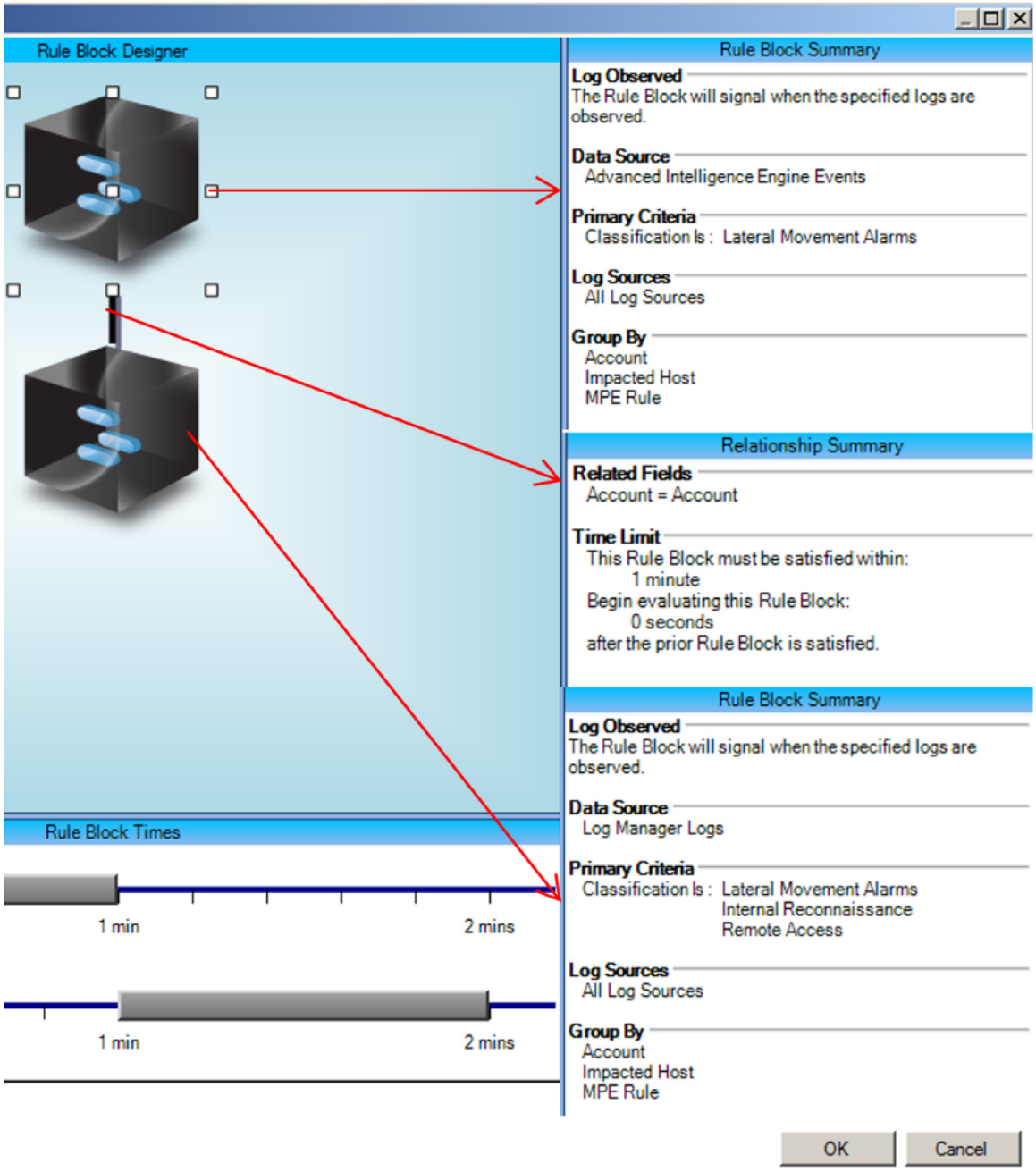

Fig. 19. Correlation rule for log aggregation via subplane classification.

where sophisticated attack detection required a series of multi block rules to account for all possible indicators of compromise; for instance one multi block rule for administrator logon events with Windows event 4672 and another multi block rule for administrator logon events detected by a whitelist/blacklist. Expanded classification labels allowed for one rule to be created to label any logs observed with one of several characteristics of an administrator logon as privilege escalation activity. This rule could include the presence of a Windows 4672 log, a specific account name on a whitelist/blacklist, the system account running a user process, additions to certain security groups, a network device known vendor/default account etc.

Once all indicators of "privilege escalation" activity were assigned the appropriate label by the first log classification rule, a second rule could then search for any "privilege escalation" labeled events within a certain time frame, or suspicious subplan activity that would not normally generate alarms in isolation. This simple two stage construction method effectively replaced the need to make an exhaustive and convoluted series of every possible permutation of privilege escalation events within the SIEM. Furthermore, this labeling and greedy query approach provided an elegant solution to aggregating similar events, with compatible meta data, was not dependent upon strict event sequencing, nor was it susceptible to false negatives due to system time drift.

Fig. 18 depicts a modified SIEM correlation rule being created to detect processes associated with internal reconnaissance activity. The newly created classification label of "internal reconnaissance" will be applied to the newly created event. Multiple meta data fields will be populated within the newly created event for future correlation such as: account name, event rule title (stored

as MPE rule), data source name, process name, command arguments (stored as URL) and impacted host (the dereferenced host name and IP address resolved by the SIEM database). Fig. 19 depicts a multi block rule designed to aggregate suspicious events, such as the one created by the previous rule, with other similarly classified events if a high confidence alert is generated within the "lateral movement alarm" classification.

Investigation framework analysis The pool of resident security analysts at the MSSP was used to evaluate the efficacy of the novel framework during investigations. Benefits noted from this evaluation include:

- Identification of potential false negatives due to data omission or errors in programmatic SIEM correlation logic. Newly created multistage rules were less likely to fail to generate alarms based on missing data, poorly configured data sources, or improperly configured "group by" meta data fields.
- Improved communication between analysts, SIEM engineers and stakeholders. The new framework provided a mechanism for describing and contextualizing alerts generated by the SIEM in a manner that was shared across the entire SIEM user base; analysts, engineers and clients. Additionally, the phased structure of the framework allowed for predictive analysis of potential preceding or expected future events based on observed alerts.
- Operational process efficiency gains due to reduction in redundant queries. More descriptive alert names and aggregate data contained within alarms provided analysts with additional information useful in developing recommendations to clients. Additionally, clients were less likely to request additional information from analysts, thus reducing the requirement for manual queries or investigations.

7. Conclusions

The modified ontology appears to be an improvement over the baseline SIEM ontology in every dimension measured in this paper. The modifications resulted in a drastic reduction in the number of alerts that provide little forensic value to analysts. Additionally, the amount of data provided on a per alert basis was greatly improved through the novel aggregation mechanism of pairing the modified log ontology classification labels with identity metadata fields specific to each kill-chain phase. Though the primary motivation for the modified log ontology revolved around alert forensic value, marked improvements in SIEM resource consumption were noted following the implementation of simplified correlation rule queries. Additionally, it is assessed that simpler correlation queries will result in decreased administrative effort to maintain the SIEM system. These improvements are assessed to have improved the mean time required to detect security events based on the following factors:

- Increased visibility during network security attacks through improved detection rate (roughly 70% improvement in number of test cases detected).
- Increased number of metadata fields contained within alerts generated.
- Decreased total alert volume.
- Decreased effort required by engineers to deploy detection rules.
- Decreased system resource requirements preventing potential processing bottlenecks.

8. Compliance with ethical standards

The research work reported here was conducted by the authors at a research institution. There are no conflicts of interest. The authors are researchers and the work report here is entirely the result of their research interests. No financial support was provided for the work by any entities. No human or animal subjects were used in the study. No individuals (other than the authors) were involved in the research work and no personal or confidential information is reported in the work.

Declaration of Competing Interest

None.

References

Aminanto, M.E., Zhu, L., Ban, T., Isawa, R., Takahashi, T., Inoue, D., 2019. Automated threat-alert screening for battling alert fatigue with temporal isolation forest. In: Proceedings of the 17th International Conference on Privacy, Security and Trust (PST), pp. 1–3. doi:10.1109/PST47121.2019.8949029.

Bryant, B.D., Saiedian, H., 2017. A novel kill-chain framework for remote security log analysis with SIEM software. Comput. Secur. 67, 198–210. doi:10.1016/j.cose.2017.03.003.

Chien, S., Chang, E., Yu, C., Ho, C., 2007. Attack Subplan-based attack scenario correlation. In: Proceedings of the International Conference on Machine Learning and Cybernetics, 4, pp. 1881–1887. doi:10.1109/ICMLC.2007.4370455.

Denning, D.E., 1987. An intrusion-detection model. IEEE Trans. Softw. Eng. SE-13 (2), 222–232. doi:10.1109/TSE.1987.232894.

Denning, D.E., 2001. Activism, Hacktivism, and Cyberterrorism: the internet as a tool for influencing foreign policy. In: Arquilla, J., Ronfeldt, D. (Eds.), Networks and Netwars. RAND, pp. 239–288.

Eric Hutchins, M.C., Amin, R., 2011. Intelligence-driven computer network defense informed by analysis of adversary campaigns and intrusion kill chains. In: Ryan, J. (Ed.), Leading Issues in Information Warfare and Security Research, 1. Academic Publishing International Limited, pp. 80–106.

FireEye, 2013. Apt1: Exposing one of China's cyber espionage units. https://www.fireeye.com/content/dam/fireeye-www/services/pdfs/mandiant-apt1-report.pdf. Accessed: 2020-01-14.

Flynn, J., 2012. Intrusion along the kill chain. Proceedings of the Blackhat Security Conference.

Garcia, M., Neves, N., Bessani, A., 2018. Sieveq: a layered BFT protection system for critical services. IEEE Trans. Dependable Secure Comput. 15 (3), 511–525. doi:10.1109/TDSC.2016.2593442.

Garcia-Teodoro, P., Diaz-Verdejo, J., Macia-Fernandez, G., Vazquez, E., 2009. Anomaly-based network intrusion detection: techniques, systems and challenges. Comput. Secur. 28 (1), 18–28. doi:10.1016/j.cose.2008.08.003.

Grano, J.J., Webster, W., Ostergaard, D., Romney, M., Cohen, J., Miron, M., 2005. Intelligence and information sharing initiative: homeland security intelligence and information fusion. US Department of Homeland Security, Washington, DC.

Hald, S.L.N., Pedersen, J.M., 2012. An updated taxonomy for characterizing hackers according to their threat properties. In: Proceedings of the 14th International Conference on Advanced Communication Technology (ICACT), pp. 81–86.

Kim, I., Oh, D., Yoon, M.K., Yi, K., Ro, W.W., 2013. A distributed signature detection method for detecting intrusions in sensor systems. Sensors 13 (4), 3998–4016. doi:10.3390/s130403998.

Legrand, V., State, R., Paffumi, L., 2008. A dangerousness-based investigation model for security event management. In: Proceedings of the Third International Conference on Internet Monitoring and Protection, pp. 109–118. doi:10.1109/ICIMP.2008.16.

Madani, A., Rezayi, S., Gharaee, H., 2011. Log management comprehensive architecture in security operation center (SOC). In: Proceedings of the International Conference on Computational Aspects of Social Networks (CASoN), pp. 284–289. doi:10.1109/CASON.2011.6085959.

MITRE,.bibinfoyear2014 Mitre att&ck matrix for enterprise. https://attack.mitre.org. Accessed: 2020-01-14.

Novikova, E., Kotenko, I., 2013. Analytical visualization techniques for security information and event management. In: Proceedings of the 21st Euromicro International Conference on Parallel, Distributed, and Network-Based Processing, pp. 519–525. doi:10.1109/PDP.2013.84.

Safarzadeh, M., Gharaee, H., Panahi, A., 2019. A novel and comprehensive evaluation methodology for SIEM. In: Proceedings of the 15th International Conference on Information Security Practice and Experience (ISPEC), pp. 476–488. doi:10.1007/978-3-030-34339-2_28.

Silowash, G., Lewellen, T., Burns, J., Costa, D., 2013. Detecting and Preventing Data Exfiltration Through Encrypted Web Sessions via Traffic Inspection. Technical Report. Software Engineering Institute, Carnegie Mellon University, Pittsburgh, PA.

Strom, B.E., Applebaun, A., Miller, D.P., Nickles, K.C., Pennington, A.G., Thomas, C.B., 2018. MITRE ATT&CK: Design and Philosophy. Technical Report. MITRE.

Valeur, F., Vigna, G., Kruegel, C., Kemmerer, R.A., 2004. Comprehensive approach to intrusion detection alert correlation. IEEE Trans. Dependable Secure Comput. 1 (3), 146–169. doi:10.1109/TDSC.2004.21.

Williams, A., Nicollet, M., 2005. Improve IT Security With Vulnerability Management, Gartner ID Number: G00127481, Gartner, May 2005. (https://www.gartner.com/en/documents/480703/improve-it-security-with-vulnerability-management).

Zhong, C., Yen, J., Liu, P., Erbacher, R.F., 2016. Automate cybersecurity data triage by leveraging human analysts cognitive process. In: Proceedings of the IEEE 2nd International Conference on Big Data Security on Cloud (BigDataSecurity), IEEE International Conference on High Performance and Smart Computing (HPSC), and IEEE International Conference on Intelligent Data and Security (IDS), pp. 357–363. doi:10.1109/BigDataSecurity-HPSC-IDS.2016.41.

Zhong, C., Yen, J., Liu, P., Erbacher, R.F., 2019. Learning from experts' experience: toward automated cyber security data triage. IEEE Syst. J. 13 (1), 603–614. doi:10.1109/JSYST.2018.2828832.

Blake D. Bryant, MS, CISSP, MCITP, CCNA, completed his MS degree in information technology at the University of Kansas in 2016 and is currently a Ph.D. candidate in computer science there. He has 10+ years of practical cybersecurity experience in leading computer security organizations (private and military) and is currently a Professor Practice at the Department of Electicial Engineering & Computer Science at the University of Kansas (KU). Bryant is also a member of the US Army Reserves.

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 computer science 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.