Prototype Open-Source Software Stack for the Reduction of False Positives and Negatives in the Detection of Cyber Indicators of Compromise and Attack

Hybridized Log Analysis Correlation Engine and Container-Orchestration System Supplemented by Ensemble Method Voting Algorithms for Enhanced Event Correlation

Steve Chan
Decision Engineering Analysis Laboratory
San Diego, California U.S.A.
email: schan@dengineering.org

***Abstract*—A prototypical solution stack (Solution Stack #1) with chosen Open-Source Software (OSS) components for an experiment was enhanced by hybridized OSS amalgams (e.g., Suricata and Sagan; Kubernetes, Nomad, Cloudify and Helios; MineMeld and Hector) and supplemented by select modified algorithms (e.g., modified N-Input Voting Algorithm [NIVA] modules and a modified Fault Tolerant Averaging Algorithm [FTAA] module) leveraged by ensemble method machine learning. The preliminary results of the prototype solution stack (Stack #2) indicate a reduction, with regards to cyber Indicators of Compromise (IOC) and indicators of attack (IOA), of false positives by approximately 15% and false negatives by approximately 47%.**

Keywords-Threat Intelligence Processing Framework (TIPF); Security Orchestration (SO); Log [Analysis] and Correlation Engine (LCE); Container-Orchestration System (COS); Dynamic Service Discovery (DSD).

I. INTRODUCTION

At an Advanced Computing Systems Association (a.k.a. Unix Users Group or USENIX) Enigma Conference, Rob Joyce, the head of the National Security Agency's (NSA) Tailored Access Operations (TAO) hacking team noted that, "If you really want to protect your network, you have to know your network, including all the devices and technology in it." He went on to add that a successful attacker will often know networks better than the people who designed and run them [1]. The onus of Joyce's statement is very much, in contemporary times, carried by Managed Security Service Providers (MSSPs). The increasing level of cyber threats has obligated MSSPs to use a defense-in-depth methodology of layering various security appliances, and it has been noted that much of the successful commercial software applications in this arena is principally comprised of either the original or variants of Open-Source Software (OSS) projects. Interestingly, commercial offerings have, in some cases, become black boxed. The ensuing risk is that 41% of cyber-security applications contain high-risk open source vulnerabilities [2], and according to the 2018 Open Source Security and Risk Analysis (OSSRA) report by Black Duck of Synopsys, these risks are increasing. Without a firm understanding of the innards, MSSPs cannot readily ascertain the risk of the black boxed commercial offering itself. Accordingly, MSSPs are endeavoring to put forth their own offerings (also for market differentiation), in a white box fashion; for the purposes of adhering to Joyce's recommendation, the white box approach can be, in some cases, more effective than the black box approach, but it is a much more difficult pathway in terms of the sophistication needed to understand and appropriately orchestrate the various involved subsystems. By way of example, a Verizon Data Breach Report had articulated that those with robust log analysis and correlation were least likely to be a cyber victim; yet, legacy approaches to this particular challenge are often highly manual in nature, thereby creating complex workflows and extending the time needed for implementation (rather than decreasing the time needed, as desired by the MSSPs). To further the complexity, while various security appliances are quite successful at detecting and logging attacks and anomalous behavior, contemporary threats are characterized by being distributed in nature, acting in concert across varied systems, and employing advanced detection evasion techniques. Accordingly, MSSPs are turning to various means of automation, correlation, and orchestration. This paper presents preliminary findings from an experiment conducted, which focused upon comparing a prototypical solution stack with one that was enhanced by hybridized tools and supplemented by select modified algorithms.

This paper describes an experiment of clustering by class and hybridizing tools within the same class with certain decision-support accelerants to improve detection and decision-making. The paper first presents a solution stack (Solution Stack #1) with chosen OSS and then presents an enhanced solution stack (Solution Stack #2) aiming to reduce false positive and false negative of cyber alarms. It then proposes a method of leveraging inputs channeled via multiple OSS components (within the same class) for various classes and utilizes ensemble method machine learning. Solution Stack #1 is an original contribution as it combines OSS comments via glue code. Solution Stack #2 is an original contribution as it utilizes hybridized amalgams not discussed robustly elsewhere in literature or implemented as described herein. The N-Input Voting Algorithm (NIVA) and Fault Tolerant Averaging Algorithm (FTAA) algorithms

 ISBN: 978-1-61208-683-5

utilized are variants from the originals and have unique architecture and glue code to effectuate their implementation; indeed, the implementation was quite challenging. The paper is organized as follows: Section II discusses the trending toward the increasing utilization of MSPs, specifically MSSPs. The discussion also reviews the increasing level of cyber threats, which have obligated MSSPs to use a defense-in-depth methodology of layering various security appliances as well as their own differentiated solution stack offerings. Subsequently, Section III discusses the acknowledged layers of a MSSP solution stack, regardless of the diversification. The layers range from Remote Monitoring and Management to Dynamic Service Discovery. Then, Section IV delves into the predilection of OSS for the experiment of this paper and the preferred licenses, which include, among others, the classic GNU General Public License as well as the Affero General Public License. Section V discusses the OSS components utilized for the first phase of the experiment (as part of Solution Stack #1). The OSS projects discussed range from Project-Open to GOSINT. Section VI discusses various OSS amalgams, which have been hybridized for enhanced performance. Amalgams include Kubernetes, Nomad, Cloudify, and Helios. Section VII presents a posited hybridized solution stack, which includes the hybridized OSS amalgams from Section VI for the second phase of the experiment (as part of Solution Stack #2). Section VIII provides the experimental results from Solution Stack #1 (constituting Phase 1 of the experiment) as well as Solution Stack #2 (constituting Phase 2 of the experiment). In essence, the preliminary results indicate a reduction of false positives from Phase 1 of the experiment, Solution Stack #1 to Phase 2 of the experiment, Solution Stack #2 by approximately 15% and a reduction of false negatives by approximately 47%. Finally, the paper reviews and emphasizes key points in Section IX, the conclusion.

II. MANAGED SECURITY SERVICES PROVIDER TREND

According to International Data Corporation (IDC), at least 50% of the global [Gross Domestic Product] GDP will be digital by 2021 [3]. Yet, digital business has inherent cybersecurity risks, and this was articulated by the Digital Business World Congress. According to Klynveld Peat Marwick Goerdeler (KPTM), Chief Executive Officers (CEOs) view cybersecurity as their top risk and innovation challenge. As digital business (e.g., Internet of Things [IOT], Bring Your Own Device [BYOD], mobile computing, cloud computing, etc.) increases in its applications, new exploitable vulnerabilities for cybercrime will emerge. Along this vein, Cyveillance's "Cyber Intelligence Report" asserts that cybercriminals are constantly finding new ways to exploit cyber vulnerabilities [4]. Cybercrime damage costs are expected to reach USD$6 trillion annually by 2021 [5]. According to the Ponemon Institute, 98% of business respondents reported that they will spend over a million dollars in 2017 on cybersecurity; however, many of the systems and people in place are still not able to handle either simplistic or complex contemporary cyber threats [6].

According to Gartner, organizations are expected to increase spending on enterprise application software this year, shifting more of their budget to Software as a Service (SaaS), via the managed services market [7]. MarketsandMarkets asserts that the global managed services market is approximately USD$152.45 billion [8], and it is expected to grow to nearly USD$257.84 billion by 2022 [9]. According to Allied Market Research, the global market for SaaS or managed cyber security services by Managed Services Providers (MSPs) – or, more specifically, Managed Security Services Providers (MSSPs) – is expected to garner USD$40.97 billion by 2022 [10]. According to Gartner, "The EU General Data Protection Regulation (GDPR) has created renewed interest, and will drive 65 percent of data loss prevention buying decisions today through 2018," "security services will continue to be the fastest growing segment, especially IT outsourcing, consulting, and implementation services," and "by 2020, 40 percent of all managed security service (MSS) contracts will be bundled with other security services and broader IT outsourcing projects, up from 20 percent today" [11]. Between 2018-2025, "The security services segment is expected to grow at a [Compound Annual Growth Rate] CAGR of over 18%" and "the Asia Pacific is expected to be the fastest-growing region over the forecast period, …[due] … to the growing adoption of … managed services by the small and medium-sized enterprises [SMEs], which are expected to drive the market growth" [12] (albeit large enterprises are still significant as they are establishing branch offices at remote locations and outsourcing to MSPs and/or MSSPs as well).

Hiscox, a cyber insurance company, states that less than 52% of small businesses have a clearly defined cyber security strategy, 65% of small businesses have failed to act following a cyber security incident, and less than 21% of small businesses have a standalone cyber insurance policy, compared to more than half (58%) for large companies [13]. According to the Ponemon Institute, 61% of small businesses experienced a breach in 2017, and, according to the National Cyber Security Alliance, 60 percent of Small and Medium Businesses (SMBs) that suffer a cyber-attack are out of business within six months of a breach [14]. Given the risk, these SMEs or SMBs are treating their exposures more seriously. Trends are changing, and according to Allied Market Research, SMBs will spend approximately USD$11 billion on remotely managed security services as well as represent the primary driver for the global remotely managed security services market's projected growth [15]; after all, many SMBs have minimal IT staffing to handle the ever-increasing complex threats on the cyber landscape. Hence, the desire and need for MSSPs is ever-increasing.

 ISBN: 978-1-61208-683-5

III. LAYERS OF A MSSP SOLUTION STACK

The following sections discuss the universally acknowledged layers of a MSSP solution stack, regardless of the differentiation.

A. Remote Monitoring and Management (RMM)

Digital business has necessitated automation, and MSPs have strived to provide Professional Services Automation (PSA) tools to meet this need. Digital business has also required remote access (e.g., mobile phones, tablets, laptops). Likewise, MSPs have deployed RMM tools to meet this need. PSA tools are fundamental for any MSP, as they may keep track of customer information, track workflow, and generate invoices from that work. Most of this described work will involve those things performed and managed through the RMM; in essence, PSA is the tool to track the work, and RMM is the tool to help effectuate that work.

B. Machine Learning (ML) for the RMM

As the RMM is a backbone for the MSP, among other applications, Atera (a company that produces software for MSPs) CEO Gil Pekelman envisions incorporating ML to assist MSPs with tasks, such as how to program an RMM. According to Pekelman, with regards to monitoring tasks, there are "hundreds of things to choose from … how do you know what are the right things to monitor" [16]? Pekelman further notes that, in the past, such decisions were based upon the MSP's experience, intuition, and suggestions from peers [16]. Current thinking centers upon the fact that ML can enhance RMM by shifting from subjective (e.g., intuition) to objective (e.g., empirically-based logic) monitoring paradigms.

C. Intrusion Detection System (IDS) and Intrusion Prevention Systems (IPS)

For cyber security, monitoring should be a central tenet of any strategy, so a robust monitoring strategy seems axiomatic. However, as the time required to operationalize a robust paradigm is non-trivial, it is often de-prioritized. A classic example involves one of the most notable security breaches to date, which involved Equifax, a consumer credit reporting agency. More than 145.5 million people were affected by the attack [17], which exploited the Apache Struts Vulnerability (CVE-2017-5638) [18]. Notably, this attack was carried out over time and 30 malicious web shells were uploaded over the course of four months [19]. Ultimately, this catastrophic breach resulted from a failure to monitor and act upon security incidents early enough in the cyber-attack lifecycle.

Among other monitoring paradigms, IDS and IPS work by actively monitoring network traffic for unusual patterns or aberrant behavior. For example, an unusually high volume of data being directed to an external Internet Protocol (IP) address (e.g., an IP address located in a country in which the organization does not perform work) might trigger an IDS or IPS alert. The following are some general approaches.

1) Signature-Based: will monitor packets on the network and compare them against a database of signatures (i.e., attributes) from known cyber-threats (i.e., similar to an antivirus approach). The deficiency is that there will be a lag between the time a new threat is discovered in the wild and when the signature for detecting that threat is applied.

2) Anomaly-Based: will monitor network traffic and compare it against an established baseline. The baseline will reflect what constitutes normal for that network (e.g., bandwidth, protocols, ports, devices, etc.). An alert is provided when traffic that is anomalous from the baseline is detected.

3) Passive: will simply detect and alert. When anomalous network traffic is detected, an alert is provided. However, human intervention is needed to take an action.

4) Reactive: will not only detect anomalous traffic and provide an alert, but will also respond by taking pre-defined, proactive actions (e.g., blocking the user or source IP address from accessing the network, etc.).

The principal difference between IDS and IPS is that while IDS will indeed provide an alert based upon anomalous network traffic, it is typically a passive system that does not prevent or terminate activity; in contrast, IPS typically undertake action. They are broadly classified as follows:

1) Network Intrusion Detection Systems (NIDS): are placed at strategic points throughout the network to monitor traffic to and from all devices. Pragmatically, although monitoring all inbound and outbound traffic seems ideal, doing so might create a bottleneck that would impair the overall speed of the network.

2) Host Intrusion Detection Systems (HIDS): are run on individual hosts or devices on the network and monitor the inbound and outbound packets to and from the device only.

D. Unified Threat Management (UTM)

In an attempt to simplify and unify matters, UTM devices typically integrate a range of security devices, such as firewalls, gateways, and IDS/IPS into a single device or platform. The consolidation of these functions can simplify management tasks and training requirements; however, it can also create a single point of failure.

E. Security Information and Event Management (SIEM)

SIEM works differently from the UTM. Rather than replacing antivirus, firewalls, or IDS/IPS, SIEM operates in a complementary fashion with these devices to collect and correlate information from the log and event data produced

 ISBN: 978-1-61208-683-5

by the disparate systems (e.g., devices, applications) on the network. While individual devices or point applications may provide various fragments of information, the SIEM assists in assembling higher order vantage points to identify security risks, which individual devices and applications may not identify. Via this defense-in-depth methodology, the SIEM can help identify attacks during the initial stages of the cyber kill chain rather than the final stages.

1) Security Incident Indicator of Compromise (IOC)

To avoid security incidents from occurring, MSSPs are increasingly leveraging IOCs (e.g., malware, exploits, vulnerabilities, IP addresses, etc.). These IOCs are typical of the evidence left behind when a breach has occurred. Utilizing IOCs forensically constitutes a reactive posture.

2) Indicator of Attack (IOA):

In contrast, the utilization of IOAs (e.g., code execution, command and control, lateral movement) segue to a proactive stance, as cyber defenders actively hunt for early warning signs that an attack may be underway.

F. Threat Intelligence Platform (TIP) and Threat Intelligence Processing Framework (TIPF)

IOCs and IOAs are amalgamated from heterogeneous external sources (e.g., Spamhaus) by TIPs, which endeavor to aggregate, correlate, and analyze threat data from multiple sources in real-time to support defensive actions. The advantage of disparate sources is that each will have varied techniques and tools for operationalizing various compliance regimes. In turn, TIPFs can, in some cases, study IOCs and IOAs so as to capture cross-incident trends. TIPFs effectively translate collected IOCs and IOAs into actionable controls for enforcement on security devices.

G. Optimizing SIEM with Security Orchestration (SO)

The SIEM is unable to, inherently, reduce the number of false positives (i.e., if the SIEM sends thousands of false alarms every day, it becomes nearly impossible to keep pace, ascertain the alerts that matter, and respond in a timely fashion). By leveraging SO, the SIEM can focus on collecting data and correlating alerts, while SO (considered an enhancement to SIEM) actions, taken across the entire security product stack, can scale SIEM capabilities by automating tasks (e.g., IP lookups, log queries, etc.) and streamlining the alert ingestion from multiple sources (e.g., TIPs, TIPFs) so as to produce tailored response playbooks, as automated, orchestrated security responses (as well as potentially handling the investigation and remediation process), such as the following:

1) Firewall

- Proactively blocks IP addresses of recognized attacks (e.g., ransomware) and/or attackers;
- Proactively blocks newly detected attackers discovered by peers within the trusted circle;
- Automatically blocks the IP address of an attacker, a compromised device from outbound communication, etc.;

2) Network Device

- Automatically takes a snapshot image of the suspected device;
- Automatically removes or quarantines the device from the network;

3) User Account

- Automatically locks an account for a period of time;
- Automatically forces the password reset of a suspicious account.

The described automated actions assist in reducing false positives and better illuminating those alerts, which require further human investigation.

H. Log [Analysis] and Correlation Engine (LCE) as a Monitoring Strategy

The "Verizon Data Breach Report: Detective Controls by Percent of Breach Victims" highlighted the fact that 71% of the breach victims were those that relied predominantly upon System Device Logs, 30% for Intrusion Detection Systems, 20% for Automated Log Analysis, 13% for SIEM, and 11% for Log Review Process [20]. In essence, a comprehensive log review process or analysis (perhaps a combination of manual and automatic log analysis) very much minimizes cyber breaches. Indeed, per various Verizon Data Breach Reports, investigators noted that a substantive portion (e.g., 66%) of victims had sufficient evidence available within their logs to discover the breach had they been more diligent in analyzing such resources [21]. Accordingly, in addition to extrospection (e.g., TIPs and TIPFs), there should be a particular emphasis placed on introspection at the log level, such as by the SIEM and SO.

Aggregating security log data, via Vulnerability Scanners (VS), further streamlines the analysis of network vulnerabilities. In general, software security updates endeavor to address vulnerabilities; with the escalating vulnerabilities populating the cyber landscape, software update deployment velocity is increasing. To address this phenomenon, DevOps (a portmanteau of "Development" and "Operations") has surged. Among other solutions, containerization is often used in DevOps; containerization supports the ability to package application dependencies with the application itself, thereby ensuring that the application will perform in a consistent fashion wherever it is deployed; these applications can be modularized further into a collection of loosely coupled services called microservices, each in a container. Containers enable instant scale, as they take microseconds to instantiate, as contrasted to a virtual machine (VM), which can take minutes. Also, VMs generally support one application per Operating System (OS), due to potential conflicts with dependencies

 ISBN: 978-1-61208-683-5

(e.g., differing versions of external Dynamic Link Libraries [DLLs]). Virtualization has optimized IT work flow processes, via the capability of running multiple OS on a single server or system. For the discussed experiment, the container approach was utilized, as containers make it possible to deploy applications on generic VMs that do not have to be preconfigured to support the involved applications. This provides more flexibility, as the VMs can be treated generically (not specifically, as in the traditional case), thereby providing the ability to leverage any of the VMs (i.e., not just ones that are prepared to accept a specific application).

Containers are, in turn, run by a Pod, which represents a running process, as it encapsulates an Application (App) container (which contains the program code and its activity) or, in some cases, multiple containers. For a Pod that runs a single container, the Pod can be construed as a wrapper, and the Pods are the managed entity rather than the containers directly. For a Pod that encapsulates an application composed of multiple co-located containers (that are tightly coupled and form a single cohesive unit of service, such as for the case of a container serving files from a shared volume, while a separate "sidecar" container refreshes those files), the Pod serves as a wrapper for both the containers and storage resources, together, as a single manageable entity. This paradigm is shown in Figure 1.

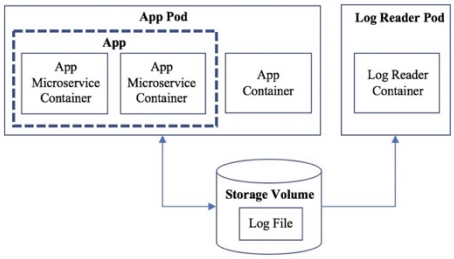

Figure 1. Exemplar Pod, Container, Volume Paradigm for Log Files

LCEs involve logging components (e.g., log reader), which can be deployed as App containers (within Pods) inside a cluster, which can refer to running an application in multiple processes (i.e., Pods), all receiving requests on the same port. In general, contemporary software applications (e.g., service-oriented architecture or SOA) are composed of multiple services; this entails multiple containers or services comprising a single App that needs to be deployed as a distributed system; such a system is complex to scale and manage. To move beyond the simple management of individual containers of simple Apps and move toward larger enterprise applications with microservices, it is necessary to utilize container-orchestration platforms.

I. Container-Orchestration System (COS)

For scalable, multi-container Apps, COS are generally utilized to automate the deployment, scaling, and management. In other words, COS will automatically start containers, scale-out containers with multiple instances per image, suspend them or shut them down as needed, and control how they access resources, such as network and data storage. Whatever the design for the COS, the task is to provide optimization for the involved container-based distributed system [22].

J. Dynamic Service Discovery (DSD)

Service discovery is a key component of most distributed systems and SOAs, as clients seek to determine the IP address port for a service that exists on multiple hosts. For a simple network, static configuration of IP addresses and ports might suffice. However, as more services are deployed, the complexity increases. For a high-performance operational system, service locations can change quite frequently as a result of automatic or manual scaling, new deployments of services, and hosts failing or being replaced; for this situation, dynamic service registration and discovery becomes much more important to avoid service interruption. Indeed, DSD is a key factor in achieving an adaptable, loosely-coupled, and more resilient SOA [23].

IV. OPEN-SOURCE PREDILECTION FOR THE EXPERIMENT

Having performed several iterative deployments of the stacks discussed herein, one experiential learning, among others, has been that past performance may not be an indicator of future results. Sometimes, commercial solutions may quickly advance to the forefront, but in some cases, many are overtaken by OSS projects. Among various reasons, innovation, particularly as pertains to the commercial offerings, may decrease after the product reaches a certain level of maturity. In several other cases, the more successful commercial solutions are comprised of either the original or variants of open-source projects. For this experiment, only OSS projects under the following licenses were utilized.

A. GNU General Public License (GPL)

GPL is a widely used free software license, which guarantees end users the freedom to run, study, share and *modify* the software.

B. MIT License (MITL)

The MIT License is another widely used free software license, which grants end users the freedom to deal with the software without restriction, including without limitation the rights to use, copy, *modify*, merge, publish, distribute, sublicense, and/or sell copies of the software.

C. Apache License

The Apache License is yet another utilized free software license that allows the user of the software the freedom to use the software for any purpose, to distribute it, to *modify* it, and to distribute modified versions of the software, under the terms of the license, without concern for royalties (of special note, Apache License Version 2.0 requires preservation of the copyright notice and disclaimer).

 ISBN: 978-1-61208-683-5

D. *Affero General Public License*

The Affero General Public License (a.k.a. Affero GPL, Affero License) is either of two distinct, though historically related, free software licenses: (1) Affero General Public License Version 1.0 (AGPLv1), which is based upon the GNU General Public License Version 2.0, and (2) Affero General Public License Version 2.0 (AGPLv2), which is a transitional license for an upgrade path from AGPLv1 to the GNU Affero General Public License, which is compatible with GNU GPL Version 3.0. Both versions of the Affero GPL were designed to close a perceived Application Service Provider (ASP) loophole in the GPL (i.e., using, but not distributing software, left copyleft provisions untriggered).

E. *Mozilla Public License*

The Mozilla Public License (MPL) defines rights as passing from "Contributors" who create or modify source code, through an auxiliary distributor (themselves a licensee), to the licensee. It grants copyright and patent licenses allowing for free use, *modification*, distribution, and exploitation of the work, but it does not grant the licensee any rights to a contributor's trademarks.

There are various solution stacks [24], but Figure 2 constitutes one exemplar and is what was utilized for the first phase of the experiment. As shown, TIPF fed its output back to the SIEM, and VS fed its output to the LCE (orange pathway).

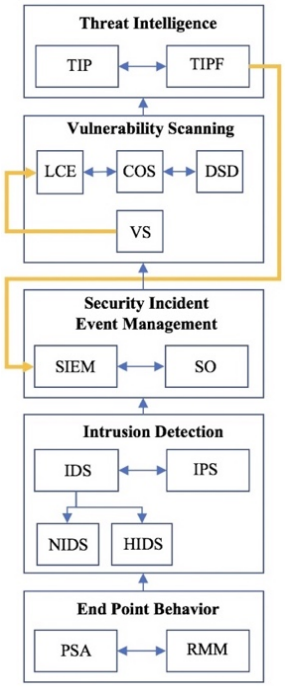

Figure 2. Exemplar Solution Stack for the First Phase of the Experiment: Solution Stack #1

V. COMPONENTS UTILIZED FOR THE EXPERIMENT

The various components utilized for the experiment included the following, which are presented in Subsections A-K.

A. *PSA*

1) Project-Open (commodity modules): free | open-source | GNU GPL Version 3.0 | PSA | application.

B. *RMM*

1) Comodo One: free | open-source | MIT License | RMM | platform. Comodo One is produced by Comodo, a cyber security company that is known for being the world's second largest Certificate Authority (CA) and was, at one time, the largest issuer of Secure Sockets Layer (SSL) certificates.

C. *IDS*

1) Security Onion: free | open-source | GNU GPL Version 2.0 | NIDS | platform.

2) OSSEC & Wazuh: free | open-source | GNU GPL Version 2.0 | HIDS | system.

3) Sagan: free | open-source | GNU GPL Version 2.0 | NIDS + HIDS | engine.

D. *IPS*

1) Suricata: free | open-source | GNU GPL Version 2.0 | IPS | engine. Suricata was developed by the Open Information Security Foundation (OISF). It is partly funded by the Department of Homeland Security's Directorate for Science and Technology and is designed to work with Snort rulesets.

E. *SIEM*

1) OSSIM: pseudo-free | open-source | GNU GPL Version 3.0 | SIEM | platform. The OSSIM project began in 2003, and in 2008, it became the basis for AlienVault. The commercial variant of OSSIM is entitled, "AlienVault Unified Security Management."

F. *SO*

1) PatrOwl: free | open-source | Affero General Public License | SO | platform.

G. *LCE*

1) Sagan: free | open-source | GNU GPL Version 2.0 | LCE | engine.

2) OpenVas: free | open-source | GNU GPL | VS | framework. OpenVAS is a member project of the Software in the Public Interest (SPI), which has hosted Wikimedia Foundation board elections and audited tallies as a neutral third party.

 ISBN: 978-1-61208-683-5

H. COS

1) Kubernetes: free | open-source | Apache 2.0 | COS | platform. It was originally designed by Google and is now maintained by the Cloud Native Computing Foundation. At the core, coordination and storage is provided by etcd.

2) Nomad: free | open-source | Mozilla Public License 2.0 | COS | tool.

3) Cloudify: free | open-source | Apache 2.0 | COS | platform. It is a software cloud and NFV orchestration product originally created by GigaSpaces Technologies (an Israeli company focused on space-based architectures [e.g., tuple spaces]), and then spun out.

4) Helios: free | open-source | Apache 2.0 | COS | platform. Spotify created Helios, which is a key component of their scalability strategy. Helios has the capacity to perceive when a "container is dead;" if a mission critical container is accidentally closed down, Helios quickly loads one back up.

I. DSD

1) Consul: free | open-source | Mozilla Public License 2.0 | DSD | tool. Consul is designed for multi-datacenter service discovery.

J. TIP

1) MineMeld: free | open-source | Apache 2.0 | TIP | platform. As part of its commitment to the security community and mission of driving a new era of threat intelligence sharing, Palo Alto Networks released MineMeld to the community-at-large.

2) HECTOR: free | open-source | GNU GPL Version 3.0 | TIP | platform. HECTOR is an open source initiative originally sponsored by the University of Pennsylvania School of Arts & Sciences (SAS).

K. TIPF

1) GOSINT: free | open-source | GNU GPL Version 3.0 | TIPF | framework. As part of its commitment to the security community and mission of driving a new era of threat intelligence sharing, CISCO release GOSINT to the community-at-large.

The aforementioned components were utilized in both the first and second phases of the experiment. Figure 3 presents the previously presented exemplar solution stack with the various components; each component is represented in accordance to its classification herein. For example, OSSEC & Wazuh would be C-2, Suricata would be D-1, OSSIM would be E-1, Kubernetes would be H-1, and MineMeld would be J-1.

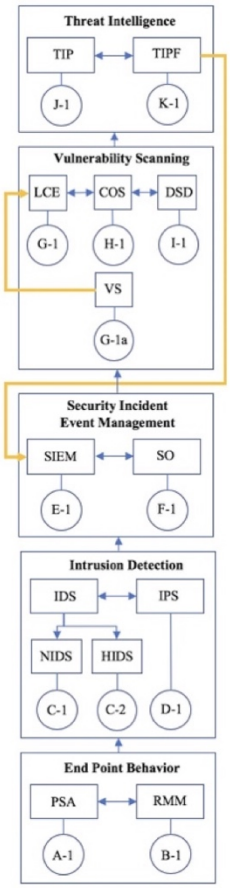

Figure 3. Exemplar Solution Stack with specified Components for the First Phase of the Experiment: Solution Stack #1

The experiment leveraged the open-source Elasticsearch, Logstash, Kibana (ELK) stack for supporting certain functionality. Logstash is a server-side data processing pipeline, which ingests data from multiple sources simultaneously, transforms it, and then sends it to a search and analytics engine, such as Elasticsearch. Kibana supports visualization analytics within Elasticsearch.

VI. HYBRIDIZING FOR ENHANCED PERFORMANCE

Preliminary results from the exemplar solution stack were obtained. It was posited that the results could be improved by enhancing the exemplar solution stack with complementary tools and supplemented by select modified algorithms. The hybridizations are discussed below.

A. Suricata and Sagan

Although Snort may be the world's most deployed IPS, its current limitation is that it, for all intents and purposes, is fundamentally single-threaded; hence, it does not take advantage of multi-core machines without special configurations. Furthermore, results show that a single instance of Suricata is able to deliver substantially higher performance than a corresponding single instance of Snort or multi-instance Snort [25]. However, Sagan utilizes a

 ISBN: 978-1-61208-683-5

multi-threaded architecture and encompasses both NIDS and HIDS while Snort and Suricata are just NIDS [26]. In brief, Sagan was utilized to complement Suricata.

B. *Kubernetes, Nomad, Cloudify and Helios*

While Kubernetes is specifically focused on Docker, Nomad is more general purpose. Nomad supports virtualized, containerized, and standalone applications. Kubernetes is wrapped by Application Programming Interface (API) controllers, which are consumed by other services that, in turn, provide higher level APIs for features (e.g., scheduling). Kubernetes documentation states that it can support clusters greater than 5,000 nodes and can support a Multi-Availability Zone (AZ)/multi-region configuration; however, Nomad has operationally proven to scale to cluster sizes that exceed 10,000 nodes in real-world production environments [27]; Nomad is designed to be a global-scale scheduler and natively supports multi-datacenter and multi-region configurations [27]. Cloudify is quite good at hybrid cloud deployment, and Helios is very good at removing single points of failures, as "numerous Helios-master services can react … at the same time" [28]. In brief, Kubernetes, Nomad, Cloudify and Helios were utilized together as a hybridized amalgam.

C. *MineMeld and Hector*

The consolidation and correlation functions performed by MineMeld can be nicely complemented, via HECTOR, which allows for correlation between otherwise unrelated security data points and metrics to extrapolate context. In brief, Hector was utilized to complement MineMeld.

VII. POSITED HYBRIDIZED SOLUTION STACK

The prototypical exemplar solution stack with the specified components for the experiment was as delineated in Section V, Figure 3. The solution stack was revised to include the hybridized groupings of Section VI. Of the 18 components included, the sponsor organizations self-described these components as: an application (1), system (1), tool (2), framework (2), engine (3), and platform (9). The revised, prototype solution stack with hybridized groupings and modified algorithms for ensemble ML is shown in Figure 4. As can be seen in Figure 4, each set of groupings passed their outputs to modified N-Input Voting Algorithm (NIVA) modules [31], which acted in concert with a modified Fault Tolerant Averaging Algorithm (FTAA) module [32], via ensemble method ML. For Intrusion Detection, C-1, C-2, and C-3 passed their outputs to NIVA-1, whose output was refined by FTAA and the resultant was N-1 (red pathway). For Vulnerability Scanning, H-1, H-2, H-3, and H-4 passed their outputs to NIVA-2, whose output was refined by FTAA and the resultant was N-2 (red pathway). For Threat Intelligence, J-1 and J-2 passed their outputs to NIVA-3, whose output was refined by FTAA and the resultant was N-3 (red pathway).

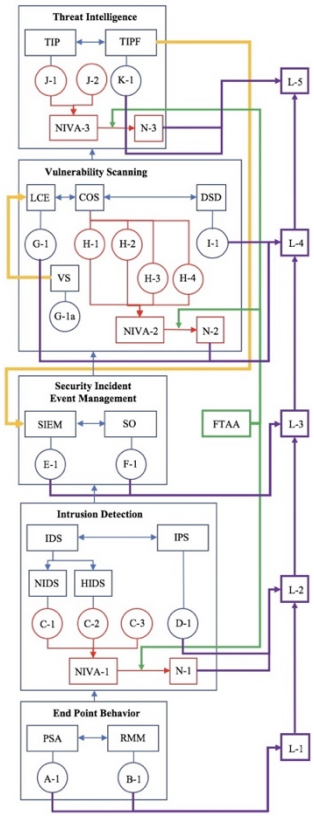

Figure 4. Revised, Prototype Solution Stack with specified Hybridized Components for the Experiment and NIVA, FTAA mechanisms: Solution Stack #2

The FTAA refinement pathways are illuminated (green pathway). The various interim steps were as follows: (A-1)&(B-1)->(L-1), (N-1)&(D-1)->(L-2), (E-1)&(F-1)->(L-3), (G-1)&(N-2)&(I-1)->(L-4), and (K-1)&(N-3)->(L-5). Each layer of the solution stack passed its output to the layer above; hence, End Point Behavior (L-1) -> Intrusion Detection (L-2) -> Security Incident Event Management (L-3) -> Vulnerability Scanning (L-4) -> Threat Intelligence (L-5) (purple pathway). Of course, the TIPF fed its output back to the SIEM, and the VS repertoire fed its output to the LCE (orange pathway).

VIII. EXPERIMENTAL RESULTS

Two separate cyber testbeds on a single cyber range were utilized to conduct the experiment; for the purposes of this paper, the results from the two testbeds were combined and are presented together. The preliminary results, as shown in Figure 5, indicate a reduction of false positives from Phase 1 of the Experiment (Solution Stack #1) to Phase 2 of the Experiment (Solution Stack #2) by approximately 15% (from 82% to 67%) and a reduction of false negatives by approximately 47% (from 78% to 31%).

 ISBN: 978-1-61208-683-5

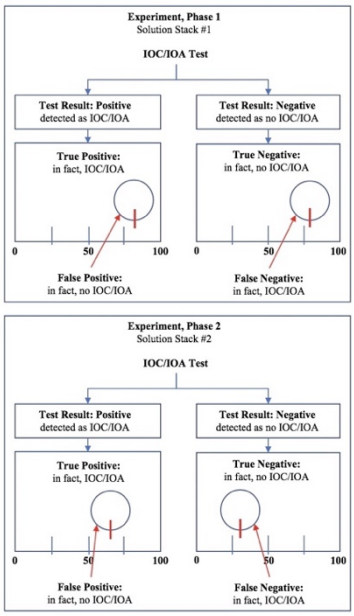

Figure 5. Results of the Experiment, Phase 1 & 2. From Solution Stack #1 to Solution Stack #2, the False Positive and False Negative rates have decreased.

For the experiment, Figure 5 was also recast, so as to be verified, with common performance measurements that were as follows: True Positive (TP), False Positive (FP), False Negative (FN), and True Negative (TN). The False Positive Rate (FPR) was calculated as FP/(FP+TN), and the True Positive Rate (TPR) was calculated as TP/(TP+FN). The Matthews Correlation Coefficient (MCC) was utilized and is shown in (1):

$$MCC = (TP)(TN)-(FP)(FN)/D \quad (1)$$

where D is defined in (2) below:

$$D=\sqrt{E} \quad (2)$$

and where E is defined in (3) below:

$$E = (TP+FP)(TP+FN)(TN+FP)(TN+FN) \quad (3)$$

The Probability Excess (PE) formula was also utilized and is shown in (4):

$$PE = (TP/P)-(FP/N) \quad (4)$$

where P and N are defined in (5) and (6) below:

$$P = TP+FN \quad (5)$$

$$N = FP+TN \quad (6)$$

The combination of MCC and PE are commonly utilized to evaluate performance of prediction methods (e.g., determining an IOC or IOA) [33].

IX. CONCLUSION

A prototypical solution stack (Solution Stack #1) with chosen OSS components for an experiment was enhanced by hybridized amalgams (e.g., Suricata and Sagan; Kubernetes, Nomad, Cloudify and Helios; MineMeld and Hector) and supplemented by select modified algorithms (e.g., modified NIVA and FTAA variants) leveraged by ensemble method ML. The preliminary results of the prototype solution stack (Solution Stack #2) indicate a reduction, with regards to IOC and IOA, of false positives by approximately 15% (from 82% to 67%) and a reduction of false negatives by approximately 47% (from 78% to 31%).

It appears that the use of complementary components conjoined with modified NIVA and FTAAs variants, leveraged by ensemble ML, shows promise. An extensive review of the prior work related to the described components, NIVAs for fault-tolerant systems, and efficient FTAAs based upon an assortment of techniques has been conducted. Future work will involve a review of updated techniques for benchmarking purposes as well as the potential involvement of other useful algorithmic modifications. Other future work, which has already commenced, will include enhancements, such as Rudder, an open-source audit and configuration management utility, which facilitates system configuration. Also, the ELK stack will be complemented with the open-source projects Sawmill and Apollo (both released by Lozi.io) to scale the log analysis environments.

ACKNOWLEDGMENT

The author would like to thank the Decision Engineering Analysis Laboratory (DEAL) for its encouragement and motivation throughout the process of pursuing and completing this research. Without their initial and continuing assistance, as well as the ideas, feedback, suggestions, guidance, resources, and contacts made available through that support, much of this research would have been delayed. The author would also like to thank VT & IE^{2}SPOMTF. This is part of a paper series on enhanced event correlation. The author would like to also thank USG leadership (e.g., the CAG). In addition, the author would like to thank ICE Cyber Security for the opportunity to serve as Chair, Scientific Advisory Board. The author would further like to thank the International Academy, Research, and Industry Association (IARIA) for the constant motivation to excel as well as the opportunity to serve as a contributing IARIA Fellow within the cyber and data analytics domains, particularly in the area of mission-critical systems.

REFERENCES

[1] I. Thomson, “NSA’s Top Hacking Boss Explains How to Protect Your Network from His Attack Squads,” The Register, pp. 1-2, 28 January 2016.

[2] D. Winder, “41% of Cyber-Secuirty Apps Contain High-Risk Open Source Vulnerabilities,” SC Magazine, pp. 1-2, 15 May 2018.

[3] “Global Interconnection Index – Digital Economy Outlook,” Equinix, pp. 1-9, 2018.

[4] H. Kenyon, “Cybercriminals Find New Ways to Exploit Vulnerabilities,” SIGNAL, pp. 1-2, March 2010.

[5] S. Morgan, “Top 5 Cybersecurity Facts, Figures, and Statistics for 2018,” Cybersecurity Business Report, pp. 1-11, 23 January 2018.

[6] “Security Practices Need to Evolve in Order to Handle Complex Threats,” Help Net Security, pp. 1-4, 9 February 2017.

[7] C. Saran, “Enterprise Software Spending Set to Grow Thanks to AI and Digital Boost,” Computer Weekly, pp. 1-6, 16 January 2018.

[8] J. Cinelli, “Five Trends to Impact Managed Service Providers in 2018,” Cloud Jumper, pp. 1-11, 2018.

[9] “Managed Services Market 2017 – Global Forecast to 2022 – Research and Markets,” BusinessWire, Berkshire Hathaway, pp. 1-3, 13 September 2017.

[10] N. Rajput, “Managed Security Services Market by Deployment Mode (Hosted or cloud-based MSS and On-premise or customer-premise equipment) and Application (Managed IPS and IDS, Distributed Denial of Services, UTM, SIEM, Firewall management, Endpoint Security and Others) - Global Opportunity Analysis and Industry Forecast, 2014 – 2022,” Allied Market Research, pp. 1-3, August 2016.

[11] “Gartner Says Worldwide Information Security Spending Will Grow 7 Percent to Reach $86.4 Billion in 2017,” Gartner, pp. 1-7, 16 August 2017.

[12] “Cloud Managed Services Market Size, Share & Trend Analysis Report By Service Type (Business, Network), By Deployment, By End-user, By Vertical, By Region, And Segment Forecasts, 2018 - 2025,” Grand View Research, pp. 1-7, April 2018.

[13] “2018 HISCO: Small Business Cyber Risk Report,” Hiscox, pp. 1-9, 2018.

[14] J. Loughlin, “Why Cyber Insurance is No Longer Optional for Restaurants,” FSR Magazine, pp. 1-7, May 2018.

[15] D. Kobialka, “MSP Research: SMBs Spend $11 Billion on Managed Security Services,” MSSP Alert, pp. 1-3, 16 August 2018.

[16] A. Brown, “Machine Learning Gains Momentum in MSP Space,” Channel Futures, pp. 1-2, 3 Feburary 2017.

[17] S. Cowley, “2.5 Million More People Potentially Exposed in Equifax Breach,” The New York Times, pp. 1-2, 2 October 2017.

[18] “CVE-2017-5638,” National Vulnerability Database, National Institute of Standards and Technology, pp. 1-5, 22 September 2017.

[19] “The Power of SIEM: Reducing Detection and Response Time in the Attack Chain,” Akamai, pp. 1-14, May 2018.

[20] W. Baker et al., “2009 Data Breach Investigations Report,” Technical Report, Verizon Business RISK Team, pp. 1-52, 2009.

[21] M. Grimaila, J. Myers, R. Mills, and G. Peterson, “Design and Analysis of a Dynamically Configured Log-based Distributed Security Event Detection Methodology,” The Journal of Defense Modeling and Simulation: Applications, Methodology, Technology, vol. 9(3), pp. 1-34, 2012.

[22] J. Ellingwood, “Architecting Applications for Kubernetes,” DigitalOcean, pp. 1-10, 20 June 2018.

[23] H. Samset and R. Braek, “Dynamic Service Discovery Using Active Lookup and Registration,” 2008 IEEE Congress on Services, pp. 545-552, 25 July 2008.

[24] “Nomad vs. Kubernetes,” Nomadproject.io, HashiCorp, pp. 1-2, 2018.

[25] J. White, T. Fitzsimmons, and J. Matthews, “Quantitative Analysis of Intrusion Detection Systems: Snort and Suricata,” Proceedings of SPIE - The International Society for Optical Engineering,” pp. 1-13, 2013.

[26] S. Cooper, “10 Top Intrusion Detection Tools for 2018,” Comparitech, pp. 1-30, 29 June 2018.

[27] S. Suhy, “NetWatcher Managed Detection and Response (MDR) Cyber Security Service,” NetWatcher, pp. 1-2, 11 August 2017.

[28] “9 Best Orchestration Open Source Docker Tools,” CodeCondo, pp. 1-8, 19 May 2018.

[29] A. Draffin, “Methodology Vs Framework – Why Waterfall and Agile Are Not Methodologies,” IT Strategies, pp. 1-2, 7 April 2010.

[30] A. Bridgwater, “What’s the Difference Between a Software Product and a Platform?” Forbes, pp. 1-2, 17 March 2015.

[31] A. Karimi, F. Zarafshan, and A. Ramli, “A Novel N-Input Voting Algorithm for X-by-Wire Fault-Tolerant Systems,” The Scientific World Journal, pp. 1-9, 2014.

[32] S. Latif-Shabgahi, “An Integrated Voting Algorithm for Fault Tolerant System,” 2011 International Conference on Software and Computer Applications, International Proc. of Computer Science and Information Technology, vol. 9, pp. 1-17, 2011.

[33] M. Vihinen, “How to Evaluate Performance of Prediction Methods? Measures and Their Interpretation in Variation Effect Analysis,” BMC Genomics, vol. 13(4), pp. 1-16, 2012.

 ISBN: 978-1-61208-683-5