American Journal of Networks and Communications
2024, Vol. 13, No. 1, pp. 19-29
https://doi.org/10.11648/j.ajnc.20241301.12

Research Article

A Proposed Approach to Integrate Application Security Vulnerability Data with Incidence Response Systems

Santanam Kasturi[1, *], Xiaolong Li[2], Peng Li[3], John Pickard[3]

[1]Department of Technology Management, Indiana State University, Terre Haute, USA
[2]Department of Electronics and Computer Engineering, Indiana State University, Terre Haute, USA
[3]Department of Technology Systems, East Carolina University, Greenville, USA

Abstract

This paper has proposed a method to develop an attack tree, from application vulnerability data discovered through tests and scans and correlation analysis using incoming transaction requests monitored by a Web Application Firewall (WAF) tool. The attack tree shows multiple pathways for an attack to shape through vulnerability linkages and a deeper analysis of the Common Weakness Enumeration (CWE) and Common Vulnerability Exposure (CVE) mapping to individual vulnerabilities. By further relating to a parent, peer, or child CWE (including CWEs that follow another CWE and in some cases precede other CWEs) will provide more insight into the attack patterns. These patterns will reveal a multi-vulnerability, multi-application attack pattern which will be hard to visualize without data consolidation and correlation analysis. The correlation analysis tied to the test and scan data supports a vulnerability lineage starting from incoming requests to individual vulnerabilities found in the code that traces a possible attack path. This solution, if automated, can provide threat alerts and immediate focus on vulnerabilities that need to be remedied as a priority. SOAR (Security Orchestration, Automation, and Response), XSOAR (Extended Security Orchestration, Automation, and Response), SIEM (Security Information and Event Management), and XDR (Extended Detection and Response) are more constructed to suit networks, infrastructure and devices, and sensors; not meant for application security vulnerability information as collected. So, this paper makes a special case that must be made for integration of application security information as part of threat intelligence, and threat and incident response systems.

Keywords

Incidence Response, Vulnerability Correlation, Attack Surface, MITRE Enterprise ATT&CK Matrix, Threat Model, Attack Tree

1. Introduction

When it comes to security and compliance, organizations must keep track of a vast array of data sources and threats. This is where SOAR (Security Orchestration, Automation, and Response), XSOAR (Extended Security Orchestration, Automation, and Response), SIEM (Security Information and Event Management), and XDR (Extended Detection and Response) come into play. SIEM is a technology that is used to collect, store, and analyze security data from various sources. The data is then used to detect and respond to security threats. SIEM technology can be used with other security

*Corresponding author: skasturi@sycamores.indstate.edu (Santanam Kasturi)

Received: 5 February 2024; **Accepted:** 22 February 2024; **Published:** 7 March 2024

solutions, such as firewalls, antivirus software, and intrusion detection systems for detecting and alerting anomalies. SOAR on the other hand is a system for automating, orchestrating, and responding to security threats after collecting data from same sources that the SIEM gathers. These responses can be customized to specific threats. If the data is not primed for an immediate response, it can be used for further analysis to make a more accurate response. XSOAR is an extended improvement over a SOAR in that it incorporates additional features like machine learning (ML), threat intelligence (TI), and security analytics for improved automation and orchestration capabilities, with a dashboard for unified security information for better threat response. XDR combines multiple detection and response techniques like SIEM and XSOAR into a single platform, combining data from different sources (network traffic, endpoint logs, and threat intelligence) to provide a more comprehensive picture of security events. Security teams gain speed through this technology integration in responding to threats by getting a full view of the security posture. XDR is not a replacement to SIEM or XSOAR but complements them.

A survey of the application of SIEM/XSOAR/XDR technologies in a variety of industries is discussed briefly. The literature cited starts with challenges identified in SIEM technologies and then covers the latest review on the state of SIEM [1, 2]. The industries and domains covered are Air Traffic Control (ATC) [1], Intrusion Detection Systems (IDS) [3], Industrial Control Systems (ICS) and critical infrastructures [4, 5], and Wind Energy Systems [6]. This literature study gives a good sweep of systems that are not just IT, but also includes application of SIEM technology outside of IT. In addition to SIEM, the literature also includes scope of XSOAR, EDR, and XDR. Tables 1 - 4 below gives a comprehensive summary of SIEM, SOAR/XSOAR, EDR, and XDR [7, 8].

Table 1. *Security Information and Event Management (SIEM).*

Characteristics	Advantages	Challenges	Deployment
Collect security event logs and telemetry data in real time for threat detection and compliance use cases, analyse data in real time, analyse incidents and impact, report, store logs and relevant information.	Irreplaceable, provides a holistic view, establishes a threshold for critical control, provides a data lineage to trace back to the initial attack and ensures hardening of the security posture and provides a sure means for real-time analysis. Supports audit and mandatory regulatory requirements in managing threats.	Vulnerable to attacker countermeasures, expensive to deploy and maintain, correlation to attack source and target is a challenge, can generate a lot of alerts and false positives, requires skilled analysts, attack variants can pose problems, a new and evolving market with too many players, scalability is a big factor and has a multiplier effect and with an ever-growing number of connected devices and assets leading to alert fatigue.	Network, devices, sensors, and all infrastructure components leading to event collection; event normalization; set action rules for protect, remove, and respond; event storage and monitoring.

Table 2. *Endpoint Detection and Response (EDR).*

Characteristics	Advantages	Challenges	Deployment
Endpoint monitoring and collecting potential threat activity, analysing data, and identifying malicious patterns, provide automatic response with action to stop or remove threats with alerts.	EDR's can provide critical context to detect advanced threats, can run automated response activity to isolating an endpoint from the network in real-time, to stop and prevent further spreading of the issue.	Standardization of unstructured data; setting correlation rules are business specific and depends on analyst knowledge; behaviour analysis is based on knowledge of system, environment, and a-priori knowledge; requires extensive ground truth data to create behaviour rules.	Resources located on the endpoint like collected events, logs, and binaries, are correlated, and analysed to determine whether a suspicious activity occurs on the host, are signature-based solutions for pattern recognition. Does not cover networks.

Table 3. *Extended / Security Orchestration, Automation and Response (X/SOAR).*

Characteristics	Advantages	Challenges	Deployment
X / SOAR systems (Extended / Security Orchestration, Automation, and response can be used to	Plays together with SIEM, uses threat and telemetry data across a wide range of target monitoring	Playbooks can become repetitive and if not reviewed can become obsolete. Although they save a lot	Host based intrusion detection systems (HIDS) and Network based intrusion detection sys-

Characteristics	Advantages	Challenges	Deployment
collect data on information security events from multiple monitoring targets, process them, and configure an automated response using typical response scenarios, can be run as a playbook.	systems to identify events and alert threats. Automating routine and repeatable incident response tasks and workflows	of time in response actions, they are created by humans from experience and a-priori knowledge of systems. As systems and attacks change, playbooks need to be revisited.	tem (NIDS) are used to create playbooks for responses based on what is detected. Streamlines incident response through an interface.

Table 4. *Extended Detection and Response (XDR).*

Characteristics	Advantages	Challenges	Deployment
Extended Detection and Response is a technology for threat detection and response, it unifies security products for detection and response and threat intelligence into a single platform.	Uses machine learning, correlation, and analytics capabilities to enhance the response time and the efficiency of the security teams. While SIEM creates alerts, XDR does a deeper analysis using Artificial Intelligence / Machine Learning (AI/ML) for a better representation of the threat.	It is a new concept; the technology is evolving and has not yet matured.	Covers endpoints, servers, emails, cloud, and networks.

2. Monitoring: Application Security vs Infrastructure Monitoring

In recent times, web applications have become the primary targets of attackers with their widespread use that even attackers identified as script kiddies with little knowledge can perform very sophisticated and damaging web attacks using ready-to-use attack tools. In addition, according to the analysis performed on all vulnerabilities, the rate of critical vulnerabilities in the web application layer is approximately four times that of the network layer [9]. Yet, most literature today reports studies on network-based intrusion detection systems (NIDS) than studies on web-based attack [10].

Applications have "normal" attributes and behavior. This is the "descriptive" portion of the evolutionary state of analysis (Descriptive, Diagnostic, Predictive, Prescriptive, and Cognitive). Attributes are scripts, languages, interpreters, compilations, configurations, security artifacts, connections (to other apps or databases), and infrastructure topology. Behavior points to availability, stability, response time, throughput/volumes, utilization (hard disk, memory, CPU), load conditioned activities (batch vs. pseudo-batch), workflow, error rates, and error response activities (or lack thereof) and indicates a state of normal behavior or a state of a compromised system. Basically, the artifact that defines a workflow profile would be considered an attribute, while the actual processing statistics of a transaction through the workflow would describe a set of behaviors [11]. Monitoring can be at the infrastructure or at the application level. While network and device monitoring are in the domain of infrastructure monitoring, network traffic associated with a specific application is in the domain of application monitoring. Similarly, CPU utilization or heap size measure is in the domain of infrastructure monitoring but associating that to a business transaction from an application is in the domain of application monitoring. The challenge, however, is that monitoring tools do not provide the correlation between an infrastructure analysis system behavior to an analysis of application behavior. The reason is applications, and their transactions do not reside or are contained in one system in an enterprise that has distributed application architecture. Applications are multi-tiered and are spread across vast global geographies. An application is also not a network, device, or a host, it is a piece of code that is not a physical entity for an agent to monitor and report to an SIEM, and when there are hundreds of distributed applications and many talking to each other, and all having hundreds of weaknesses it can be visualized as a spaghetti code with trapped vulnerabilities. There is a disconnect between a security operations infrastructure and application behavior, and correlating both needs to be addressed to bring in application monitoring within the ambit of security operations and threat response using SOAR or XSOAR, or even XDR.

3. Integrating Application Security Vulnerability Information Data Platform with SIEM/XSOAR/XDR

Incident response using SIEM/XSOAR/XDR are all reactive systems, these come into play after the fact that a threat has been detected. This is where threat hunting combined with threat detection complemented with historical data on existing application vulnerabilities and technical debt plays a pivotal role in making a comprehensive system. The proposed data consolidation platform is the first step towards that as it ena-

bles creating a data warehouse producing datamarts, dashboards, and a decision support system. The data, however, becomes a static representation of vulnerabilities prevailing in the application code, and this must be used with a correlation analysis [12]. The end goal is to integrate this system into an incident response system as shown in Figure 1. The correlation analysis data can be fed into the incident response platform using STIX and TAXII to be analyzed as a threat pattern. Structured Threat Information Expression (STIX) is a standardized language that uses a JSON-based lexicon to express and share threat intelligence information in a readable and consistent format across indent response systems. Trusted Automated Exchange of Intelligence Information (TAXII) is the format through which threat intelligence data is transmitted. TAXII is a transport protocol that supports transferring STIX insights over Hyper Text Transfer Protocol Secure (HTTPS). Conversion of vulnerability data into a reliable threat indicator requires analysis and understanding, and interpretation of the vulnerability and the associated risk prior to creating a threat indicator. Once the feeds are ingested into the incident response platform, while SIEM creates alerts, XDR does a deeper analysis using AI/ML for a better representation of the threat, and X/SOAR runs a playbook for automated response.

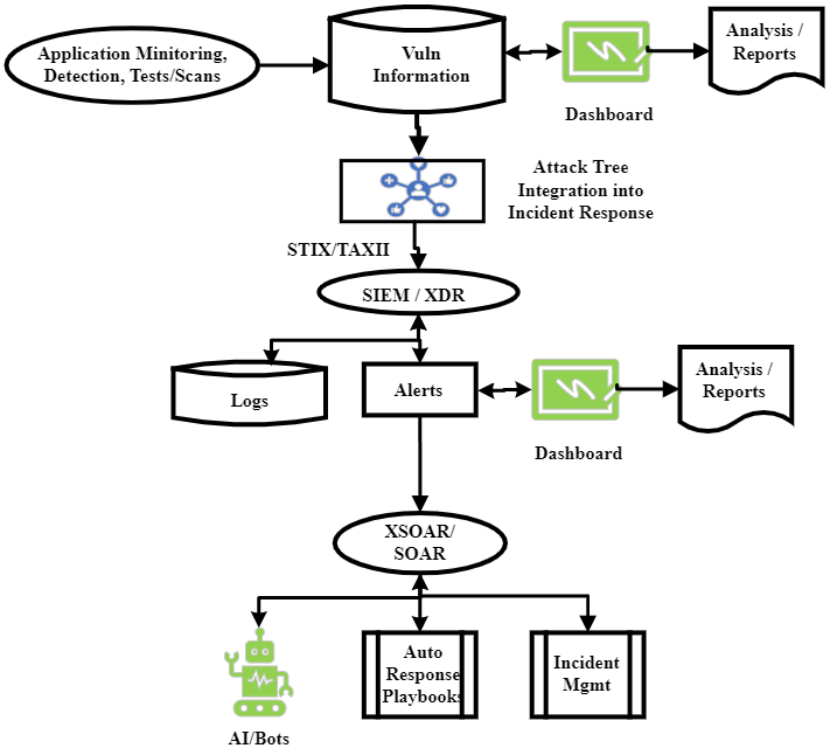

Figure 1. *Conceptual Integration of Vulnerability Data Platform and Incident Response Platform.*

The key to integration of application security vulnerability data into a threat intelligence and incident response system is integrating the attack tree. Before building an attack tree a step-by-step approach to organize and consolidate the vulnerability information and performing a correlation analysis is needed which helps in subsequently identifying the CWE's and CVE's to be mapped into an attack tree [12].

1. Identify the applications that will be under observation – usually internet facing applications that have a good business volume and are business critical.
2. Look at the vulnerabilities currently existing in these applications that are yet to be remediated from all types of internal testing that covers Static Analysis Security Testing (SAST), Software Composition Analysis (SCA), Dynamic Analysis Security Testing (DAST) and Pen testing, also known by Application Ethical Hack (AEH).
3. Identify and pick the attack types in the form of requests that shows a good transaction volume. This can be easily viewed by turning on the WAF for the applications under study.
4. Identify the vulnerabilities that match with the attack types.
5. Perform statistical and correlation analysis as discussed in an earlier work in this series [12].

A vulnerability profile can be developed from the CWEs (SAST, DAST, and AEH tests / scans) and the CVEs (from SCA) discovered using monitoring and detection methods through the parent-peer-child relationship of CWEs and CVEs and project an application-to-application spread of an attack as shown in Tables 2, 3 and 4. This can be further developed into a meaningful attack tree based on the application profile, equating to a threat model. Using the results from [12] for Cross-Site Scripting (XSS), SQL

Injection (SQLI), and Command Execution vulnerabilities found as most suitable from the correlation analysis, only for showing a representation of the linking of CWEs and CVEs for the purposes of highlighting how multi-application testing and monitoring can provide a view that vulnerabilities are not only correlated, but also linked [13-15]. The CWEs included belong in the latest 2021-2023 OWASP Top 10 and CWE Top 25 most dangerous software weaknesses (*) in each case. The three types of vulnerabilities categories XSS, SQLI, and Command Execution, as identified in CWE listing (there are many inter-linked CWE's that have a flavor of XSS, SQLI, and Command Execution), were picked for the purposes of discussion. The exercise is to build a first level vulnerability tree (or attack tree), Figure 1. A "base template" from Tables 2, 3, and 4 can then be used to create a generic tree, Figure 1 from the base template, built using textbook definition of XSS, SQLI, and Command Execution vulnerabilities as defined in National Institute of Standards and Technology (NIST) and National Vulnerability Database (NVD) CWE/CVE listings [13-15]. The next step is to then overlay the actual vulnerabilities found in the code and build a second level tree, Figure 2.

Table 5. *CWE and CVE relationships for XSS vulnerabilities and request type.*

CWE	Parent of CWE	Child of CWE	Can precede a CWE	Can follow a CWE	Is a Member of CWE*	Is a peer of CWE / Can also be	CVE Mapping
79	80, 81, 83, 84, 85, 86, 87	74	494	113, 184	1337, 1347, 1387, 1425	352, 494	CVE-2022-28599

Table 6. *CWE and CVE relationships for SQLI vulnerabilities and request type.*

CWE	Parent of CWE	Child of CWE	Can precede a CWE	Can follow a CWE	Is a Member of CWE*	Is a peer of CWE / Can also be	CVE Mapping
89	564	943, 74		456	1337, 1347, 1387, 1425		CVE-2021-43408

Table 7. *CWE and CVE relationships for Command Execution vulnerabilities and request type.*

CWE	Parent of CWE	Child of CWE	Can precede a CWE	Can follow a CWE	Is a Member of CWE*	Is a peer of CWE / Can also be	CVE Mapping
77	78, 88, 624, 917	74			1337, 1347, 1387, 1425		CVE-2022-26085
78		77		184	1337, 1347, 1387, 1425	88	CVE-2022-26085
94	95, 96, 1336	913, 74		98	1347, 1387, 1425		CVE-2023-22506

4. Developing an Attack Tree

An attack tree is developed traditionally using a threat model [16, 17]. Threat modeling allows cybersecurity professionals to assess their security posture and evaluate potential risks to an organization's assets. In the process it gives system professionals to look at all weakness and vulnerabilities in their system. This also helps in developing an attack tree, a traceable path of an attack that can be used by a threat actor in the event of an attack and helps system security professionals to monitor and detect all vulnerabilities along the attack tree. Any existing and exposed weakness can be either remedied or adequately protected in case remediation takes time. A similar concept can be used from known vulnerabilities and correlations established with incoming transaction requests monitored using a WAF and develop an attack tree. Any transaction that passes through the tree will intersect with the vulnerabilities along that path and has the potential for an attacker to exploit those weaknesses in the application code. The concept that is being proposed here is like the traditional attack tree that provides all the asset list (asset inventory) that forms an attack surface, and all the risks along the way for each asset with a known weakness. These could be servers missing security patches, network ports open without controls, or even devices that are unprotected. In this case the vulnerability-based attack tree walks through an application code that starts with a web interface and a URL, then runs down into the application code, to the component level all along touching all the vulnerabilities that are still open.

4.1. Significance of the Attack Tree

Enterprise application topologies are distributed and connected with other applications, services, and databases within a business domain. Many open-source components are shared among applications for ease of use and minimize development time. MITRE ATT&CK is a globally accessible knowledge base of adversary tactics and techniques based on real-world observations that can be used for developing threat models specific to each domain, depending on the asset topology and attack surface [18]. In its MITRE ATT&CK Enterprise Matrix, 266 techniques are listed as attack tactics, of which twelve are identified for discussion [18]. These twelve include initial access, execution, persistence, privilege escalation, defense evasion, credential access, discovery, lateral movement, collection, command and control, exfiltration, and impact. Web applications have become target of attacks in recent times, and this is a growing issue in cybersecurity. Thus, initial access is quite common, and then once in discovery phase when many vulnerabilities are found within an application, the attacker tries to make a lateral move as applications are linked by business functions and shared code. Vulnerabilities existing in one can be linked to those found in other applications. It can be visualized that a vulnerability spread across application topology can transmit an attack and spread it laterally. In this conceptual presentation here, the approach proposed shows linkages between vulnerabilities in application code and open-source component. The attack tree presented in Figure 2 is based on observations in eight applications, five of which were only considered for a deeper analysis that had significant transaction requests seen by the WAF tool [12].

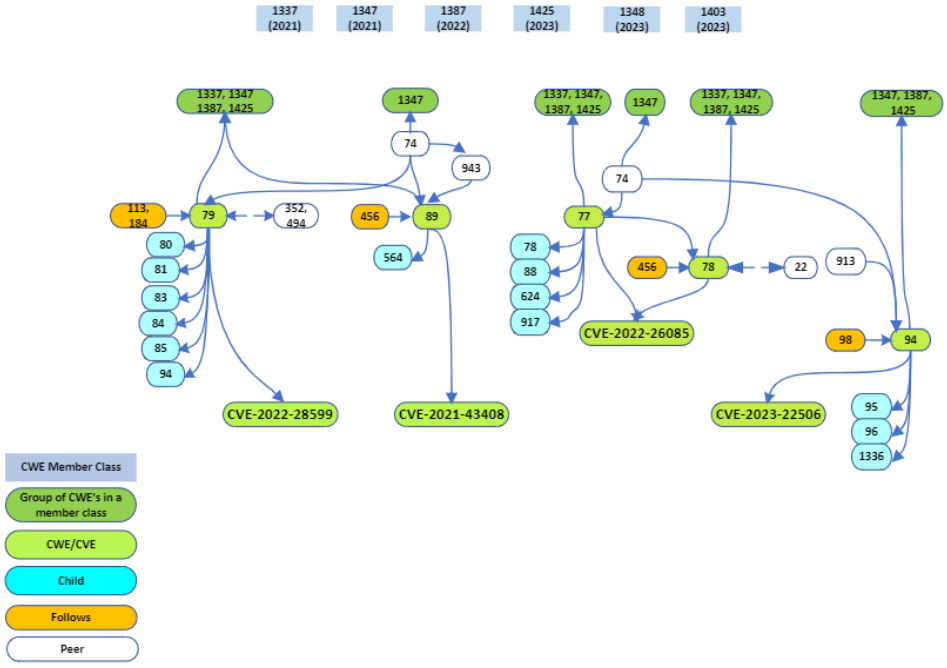

Figure 2. *Conceptual Attack Tree developed from CWE and CVE data associated with XSS, SQLI, and Command Exec vulnerabilities.*

4.2. First Level Attack Tree

From Tables 5, 6, and 7 a first level attack tree can be developed, Figure 2, it is called an attack tree for the simple reason that as the transaction passes through the code, it encounters the vulnerabilities listed in Tables 5, 6, and 7.

4.3. Second Level Attack Tree

Once the first level attack tree as shown in Figure 2 is created, the next step is to overlay real vulnerability data from the applications in consideration in a prior work [12]. The methodology for correlation analysis is to identify the transactions monitored by WAF for XSS, SQLI, and Command Execution requests. Many of the transactions are blocked based on their known digital signatures identified as malicious. The rest of the transactions are marked as valid requests and are allowed to pass through the WAF rules. The assumption is that all of these transactions are considered as valid as there are no known malicious signatures for these requests. Figure 3 shows a real attack tree with existing open vulnerabilities shown in Tables 8, 9, and 10.

Table 8. CWE and CVE relationships for existing open XSS vulnerabilities.

CWE	Parent of CWE	Child of CWE	Can precede a CWE	Can follow a CWE	Is a Member of CWE*	Is a peer of CWE / Can also be	CVE Mapping
79	80		494		1337, 1347, 1387, 1409, 1425	352	CVE-2022-24891

Table 9. CWE and CVE relationships for existing open SQLI vulnerabilities.

CWE	Parent of CWE	Child of CWE	Can precede a CWE	Can follow a CWE	Is a Member of CWE*	Is a peer of CWE / Can also be	CVE Mapping
22		20			1337, 1345, 1387, 1425		CVE-2022-38900, CVE-2022-23457, CVE-2021-37701, CVE-2021-37712

Table 10. CWE and CVE relationships for existing open Command Execution vulnerabilities.

CWE	Parent of CWE	Child of CWE	Can precede a CWE	Can follow a CWE	Is a Member of CWE*	Is a peer of CWE / Can also be	CVE Mapping
117					1337, 1347, 1387, 1425		CVE-2022-26085
611				184	1337, 1347, 1387, 1425		CVE-2022-40705, CVE-2021-33813
94					1347, 1387, 1425		CVE-2022-22965, CVE-2021-23337

Observations also found a path traversal vulnerability CWE 22, child of CWE 20, with associated CVE's CVE-2022-23457, CVE-2021-37701, and CVE-2021-37712 and shown in Table 11 and Figure 3. Although the authors [12] consider this as a weak correlation with incoming Traversal requests monitored by WAF, it has been included as part of the existing vulnerability to build the attack tree.

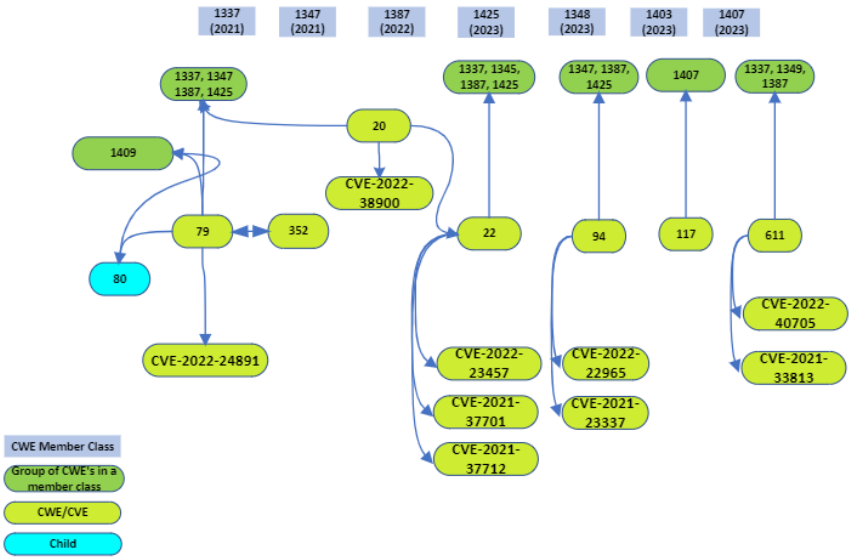

Figure 3. Real Attack Tree developed from XSS, SQLI, and Command Exec vulnerabilities found in the applications under study.

Table 11. CWE and CVE relationships for existing open Path Traversal vulnerabilities.

CWE	Parent of CWE	Child of CWE	Can precede a CWE	Can follow a CWE	Is a Member of CWE*	Is a peer of CWE / Can also be	CVE Mapping
22		20			1337, 1345, 1387, 1425		CVE-2022-23457, CVE-2021-37701, CVE-2021-37712

4.4. Creating the Threat Alert

The last step in the integration of application vulnerability data into the threat intelligence landscape and into the incident response system comprising of SIEM/XSOAR/XDR is the creation of the alert system from the attack tree analysis. The application security threat intelligence needs to complement the traditional threat intelligence systems covered by SIEM/XSOAR/XDR as in network, devices, and end point monitoring systems. This is done by employing a Web Application Firewall (WAF) / Hybrid WAF and a Runtime Application Self-Protect (RASP) combination. A web application firewall (WAF) looks at applications using HTTP traffic between the application and internet and blocks, monitors, or filter it. Typically deployed in the application layer WAF is designed to protect a web application from attacks that include SQL injection, Cross-site scripting, and session hijacking. This is a protocol layer 7 defense and not designed to defend against all types of attacks but acts as a shield in front of the application. Unlike a proxy server that protects a client machine's identity by using an intermediary, a WAF is a type of reverse-proxy, protecting the server from exposure by having requests pass through the WAF before reaching the server. A set of rules defines the operation of the WAF by filtering out what is detected as malicious traffic based on how the rules are set. Speed and ease of implementation, including quick policy updates to react to newer attacks is the biggest value a WAF provides [19, 20].

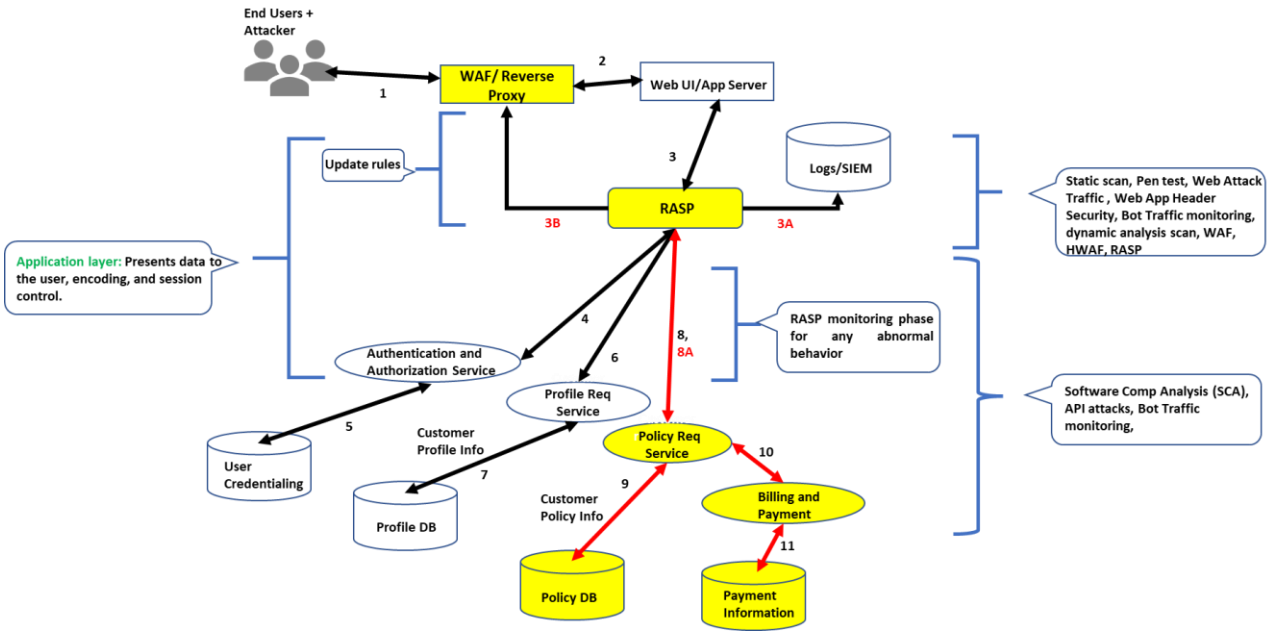

Figure 4. Topology of an attack tree of a typical web application.

A Hybrid WAF is an extension of the capability of a WAF in that it is deployed in the web server, as opposed to a WAF that is deployed in front of it. Additionally, it improves upon the traditional rules based WAF with automated detection and covers a wide range of malicious attacks by blocking them. These include SQL injection, cross-site scripting, command execution, traversal, and backdoor. To protect legacy applications, it can operate as a reverse proxy. Next-Gen WAF holds promise but still has some way to go. In the case of WAF and Hybrid WAF, a need to understand the baseline of expected traffic directed at the applications under monitoring is a key factor. One also needs visibility into which traffic

looks malicious and why. The first challenge is to know the normal traffic volumes is the organization's responsibility. The second challenge of knowing which traffic is valid and which is malicious is the tool's responsibility [21, 22]. A Run-Time Self Protection (RASP) tool will detect application code dependency and configuration level vulnerabilities in production at runtime and will help continuously to monitor, find, and block exploits. A RASP is an extension of a Hybrid WAF in the sense that a RASP goes a step ahead in following the application transaction to look for abnormalities in behavior and determine if it is an attack and not a normal business transaction. A Hybrid WAF's action stops by blocking the transaction based on known signatures [23]. An architecture for setting up a WAF-RASP threat intelligence system is proposed in [24]. While WAF blocks malicious looking transaction requests based on a database of known attack patterns and is indexed to validate every incoming transaction. Any transaction that does not find place in the database is allowed to pass through for monitoring. Here is where a vulnerability map and an attack tree based on vulnerability distribution is critical to examine through correlation analysis if the allowed transaction behaves as it should, that it will allow focus on transaction paths that intersect the vulnerability points as given by the attack tree. Figure 4 shows a schematic of how WAF/RASP combines to detect and update rules for adding more malicious patterns to known patterns based on behavior analysis. The red lines point to malicious behavior, detected, and reported by RASP, added to the logs, and then the rules are updated to block such patterns in future.

The attack tree shown in Figure 4 is a template for a typical web application, a customer facing internet application that allows access to look at products offered, get an online quote, and make an online buy. The application also allows the user to look at policy information and update billing information. The user logs into the application using a web browser session, is authenticated, and authored to access documents and payment accounts for his user profile. Assume there was an authorization error, and detected by RASP as an abnormal behavior, and the user can look at other customer information, and specifically payment information (the transactions are shown by numbers in Figure 4), then the user (an attacker) can easily obtain, through social engineering, by calling the customer service and obtain credentials to log in. This way the attacker accesses the real user's account and steals credit card information, using policy information (red lines) the user knows. Broken authentication, AP 12, is a weakness in Application Programming Interface (API's) that permits weak passwords for a user, and this can be exploited by an attacker by using credential stuffing which is another weakness associated with broken authentication AP 12 of the top 10 OWASP exploitable vulnerability [25]. The attacker can then log in to their authorized account and add another customer's credit card on file to make their account payment.

5. Conclusions

This paper has proposed an application security threat intelligence system using a vulnerability map-based attack tree that will be foundational to build a comprehensive predictive system. The paper has also proposed how application security vulnerabilities can be integrated using this attack tree with alerts ingested into SIEM, SOAR, XSOAR, and XDR to establish a fully automated attack detection and response system. As mentioned in the paper, SIEM, SOAR, XSOAR, and XDR are more constructed to suit networks, infrastructure and devices, and sensors; not meant for application security vulnerability information as collected. So, this paper makes a special case that must be made for integration of application security information as part of threat intelligence and threat response.

Abbreviations

AEH: Application Ethical Hack.
AI: Artificial Intelligence
ATC: Air Traffic Control
CVE: Common Vulnerability Exposure
CWE: Common Weakness Enumeration
DAST: Dynamic Analysis Security Testing
HIDS: Host based Intrusion Detection System
HTTPS: Hyper Text Transfer Protocol Secure
IDS: Intrusion Detection Systems
JSON: Java Script Object Notation
ML: Machine Learning
NIDS: Network based Intrusion Detection System
NIST: National Institute of Standards and Technology
NVD: National Vulnerability Database
RASP: Runtime Application Self-Protect
SAST: Static Analysis Security Testing SCA - Software Composition Analysis
SIEM: Security Information and Event Management
SCA: Software Composition Analysis
SOAR: Security Orchestration, Automation, and Response
SQLI: SQL Injection
STIX: Structured Threat Information Expression
TAXII: Trusted Automated Exchange of Intelligence Information
TI: Threat Intelligence
XDR: Extended detection and Response
XSOAR: Extended Security Orchestration, Automation, and Response
XSS: Cross-Site Scripting
WAF: Web Application Firewall

Acknowledgments

This research has been supported and guided by Dr. Xiaolong Li of Indiana State University, USA, and Dr. John Pickard and Dr. Peng Li of East Carolina University, USA.

Dr. Xiaolong Li is a professor in the Department of Electronics and Computer Engineering Technology at Indiana State University. He received his PhD in Computer Engineering from the University of Cincinnati in 2006. His primary areas of research include modeling and performance analysis of MAC protocol, Internet of Things, Wireless Ad Hoc networks, and sensor networks.

Dr. John Pickard is a professor of Information and Cybersecurity Technology at East Carolina University, North Carolina, USA. He received his PhD in Technology Management from Indiana State University in 2014. His main research areas are internet protocols, convergence of information and operations technologies, and Internet of Things applications.

Dr. Peng Li received his Ph.D. in Electrical Engineering from the University of Connecticut. His professional certifications include CISSP, RHCE and VCP. Dr. Li is currently an Associate Professor at East Carolina University. He teaches undergraduate and graduate courses in programming, computer networks, information security, web services and virtualization technologies. His research interests include virtualization, cloud computing, cybersecurity, and integration of information technology in education.

Conflicts of Interest

The authors declare no conflicts of interests.

References

[1] Cinque, M., Cotroneo, D., and Pecchia, A. Challenges and Directions in Security Information and Event Management (SIEM). In *2018 IEEE International Symposium on Software Reliability Engineering Workshops*. http://dx.doi.org/10.1109/ISSREW.2018.00-24

[2] Velásquez, J. M. L., Monterrubio, S. M. M., Luis Enrique Sánchez Crespo, L. E. S., and Rosado, D. G. Systematic review of SIEM technology: SIEM-SC birth. In *International Journal of Information Security* (2023) 22: 691–711, https://doi.org/10.1007/s10207-022-00657-9

[3] Muhammad, A. R., Sukarno, P., and Wardana, A. A. Integrated Security Information and Event Management (SIEM) with Intrusion Detection System (IDS) for Live Analysis based on Machine Learning. In *4th International Conference on Industry 4.0 and Smart Manufacturing, ScienceDirect, Procedia Computer Science* 217 (2023) 1406–1415, https://doi.org/10.1016/j.procs.2022.12.339

[4] Mern, J., Hatch, K., Silva, R., Hickert, C., Sookoor, T., and Kochenderfer, M. J. Autonomous Attack Mitigation for Industrial Control Systems. In *2022 52nd Annual IEEE/IFIP International Conference on Dependable Systems and Networks Workshops (DSN-W)*. *https://doi.org/10.48550/arXiv.2111.02445*

[5] Gonzalez-Granadillo, G., Gonzalez-Zarzosa, S., and Diaz, R. Security Information and Event Management (SIEM): Analysis, Trends, and Usage in Critical Infrastructures. In *Sensors* 2021, 21, 4759. https://doi.org/10.3390/s21144759

[6] Johnson, J., McCarty, M., Richardson, B., Rieger, C., Cooley, R., Gentle, J. P., Rothwell, B., Phillips, T., Novak, B., Culler, M., Schwalm, K., and Wright, B. Hardening Wind Energy Systems from Cyber Threats–Final Project Report. In *SANDIA REPORT*, SAND2023-12610, Printed February 2023.

[7] Nour, B., Pourzandi, M., and Debbabi, M. A Survey on Threat Hunting in Enterprise Networks. In *IEEE COMMUNICATIONS SURVEYS & TUTORIALS*, VOL. 25, NO. 4, FOURTH QUARTER 2023. https://doi.org/10.1109/COMST.2023.3299519

[8] Olteanu, I. Evaluating the response effectiveness of XDR technology in a scaled down environment. Eindhoven University of Technology, Available from: https://research.tue.nl/files/305661196/Olteanu_I.C..pdf

[9] EdgeScan. Vulnerability Statistics Report. In *Edgescan*, pp. 4-17, Available from: https://www.edgescan.com/wp-content/uploads/2019/02/edgescan-Vulnerability-Stats-Report-2019.pdf

[10] Sevri, M. and Karacan, H. Deep learning-based web application security. In *Proc. of 2nd Int. Conf. on Advanced Technologies, in Proc. Computer Engineering and Science (ICATCES)*, Antalya, Turkey, pp. 349-354, Apr. 2019.

[11] Kasturi, S. Post Implementation Evaluation of Coverage in Software Testing Using Monitoring Tools. *2020 IEEE International Conference on Computing, Power and Communication Technologies*, (GUCON), Oct 2-4, 2020, pp. 13-21, https://doi.org/10.1109/GUCON48875.2020.9231169

[12] Kasturi, S., Li, X., Pickard, J., and Li, P. Understanding Statistical Correlation of Application Security Vulnerability Data from Detection and Monitoring Tools. In 2023 33rd International Telecommunication Networks and Applications Conference, Melbourne, Australia, 2023, pp. 289-296, https://doi.org/10.1109/ITNAC59571.2023.10368476

[13] MITRE. 2022 CWE Top 25 Most Dangerous Software Weaknesses. Available from: https://cwe.mitre.org/top25/archive/2021/2021_cwe_top25.html

[14] OWASP. OWASP Top 10. *OWASP*, Available from: https://owasp.org/Top10/

[15] MITRE. Common Vulnerabilities and Exposures (CVE) Numbering Authority (CNA) Rules. *MITRE*, Available from: https://cve.mitre.org/cve/cna/CNA_Rules_v2.0.pdf; https://nvd.nist.gov/vuln

[16] Saini, V. K., Duan, Q., and Paruchuri, V. Threat Modeling Using Attack Trees. *Researchgate*, Available from: https://www.researchgate.net/publication/234738557_Threat_Modeling_Using_Attack_Trees

[17] Lohmann, P., Albuquerque, C., and Machado, R.C.S. Systematic Literature Review of Threat Modeling Concepts. In *Researchgate Conference Paper*, March 2023 https://doi.org/10.5220/0000168400003405, Available from: https://www.researchgate.net/publication/368897944_Systematic_Literature_Review_of_Threat_Modeling_Concepts

[18] Xiong, W., Legrand, E., Aberg, O., and Lagerstrom, R. Cyber security threat modeling based on the MITRE Enterprise ATT&CK Matrix. *Software and Systems Modeling (2022)* 21: 157–177 Available from: https://doi.org/10.1007/s10270-021-00898-7

[19] Akamai. Slipping Through the Security Gaps: The Rise of Application and API Attacks. *Akamai*, Available from: https://www.akamai.com/blog/security/the-rise-of-application-and-api-attacks

[20] Carielli, S., DeMartine, A., Provost, A. C. and Dostie, P. The Forrester Wave™: Web Application Firewalls, Q3 2022, The 12 Providers That Matter Most And How They Stack Up. In *Forrester*, September, Available from: https://www.forrester.com/report/the-forrester-wave-tm-web-application-firewalls-q3-2022/RES176396

[21] FASTLY. 10 Key Capabilities of the Fastly Next-Gen WAF. *FASTLY*, 2022, Available from: https://learn.fastly.com/security-10-key-capabilities-of-fastlys-next-gen-waf.html

[22] Signal Sciences. Identifying Web Attack Indicators. Available from: signal-sciences-white-paper-identifying-web-attack-indicators.pdf (signalsciences.com).

[23] Na, J. Introducing Secure Application: True Runtime Application Self-Protection (RASP) for the Modern Application. In CISCO App Dynamics. Available from*:* https://www.appdynamics.com/blog/product/application-security/

[24] Salemi, M. Automated rules generation into Web Application Firewall using Runtime Application Self-Protection. *Ecole polytechnique de Louvain*, Université catholique de Louvain, 2020. Prom.: Ramin Sadre; Legay, Axel. Available from*:* http://hdl.handle.net/2078.1/thesis:25351

[25] OWASP-API. OWASP API Security Top 10. *OWASP*, Available from: https://owasp.org/API-Security/editions/2023/en/0xa2-broken-authentication/