Fall 12-2022

A Relevance Model for Threat-Centric Ranking of Cybersecurity Vulnerabilities

Corren G. McCoy
Old Dominion University, correnm@hotmail.com

A RELEVANCE MODEL FOR THREAT-CENTRIC RANKING OF CYBERSECURITY VULNERABILITIES

by

Corren G. McCoy
B.S. August 1983, Pennsylvania State University
M.S. May 1990, Old Dominion University
M.A. May 2006, Regent University

A Dissertation Submitted to the Faculty of
Old Dominion University in Partial Fulfillment of the
Requirements for the Degree of

DOCTOR OF PHILOSOPHY

COMPUTER SCIENCE

OLD DOMINION UNIVERSITY
December 2022

Approved by:

Michele C. Weigle (Director)

Michael L. Nelson (Member)

Ross J. Gore (Member)

Faryaneh Poursardar (Member)

ABSTRACT

A RELEVANCE MODEL FOR THREAT-CENTRIC RANKING OF CYBERSECURITY VULNERABILITIES

Corren G. McCoy
Old Dominion University, 2022
Director: Dr. Michele C. Weigle

The relentless and often haphazard process of tracking and remediating vulnerabilities is a top concern for cybersecurity professionals. The key challenge they face is trying to identify a remediation scheme specific to in-house, organizational objectives. Without a strategy, the result is a patchwork of fixes applied to a tide of vulnerabilities, any one of which could be the single point of failure in an otherwise formidable defense. This means one of the biggest challenges in vulnerability management relates to prioritization. Given that so few vulnerabilities are a focus of real-world attacks, a practical remediation strategy is to identify vulnerabilities likely to be exploited and focus efforts towards remediating those vulnerabilities first. The goal of this research is to demonstrate that aggregating and synthesizing readily accessible, public data sources to provide personalized, automated recommendations that an organization can use to prioritize its vulnerability management strategy will offer significant improvements over what is currently realized using the Common Vulnerability Scoring System (CVSS). We provide a framework for vulnerability management specifically focused on mitigating threats using adversary criteria derived from MITRE ATT&CK. We identify the data mining steps needed to acquire, standardize, and integrate publicly available cyber intelligence data sets into a robust knowledge graph from which stakeholders can infer business logic related to known threats. We tested our approach by identifying vulnerabilities in academic and common software associated with six universities and four government facilities. Ranking policy performance was measured using the Normalized Discounted Cumulative Gain (nDCG). Our results show an average 71.5% to 91.3% improvement towards the identification of vulnerabilities likely to be targeted and exploited by cyber threat actors. The ROI of patching using our policies resulted in a savings in the range of 23.3% to 25.5% in annualized unit costs. Our results demonstrate the efficiency of creating knowledge graphs to link large data sets to facilitate semantic queries and create data-driven, flexible ranking policies. Additionally, our framework uses only open standards, making implementation and improvement feasible for cyber practitioners and academia.

This dissertation is dedicated to my late mother, Eliza Glover, who instilled in me the lifelong desire to learn.

ACKNOWLEDGEMENTS

I give thanks to the Lord Almighty whose grace guided me from the very beginning to the completion of this research work. Each moment during the course of this journey, I experienced the Grace of God who continuously enhanced my intelligence even during moments of despair, inspired me to move forward, opened before me unexpected avenues to guide my feet, and enlightened my thoughts with His wisdom.

My academic journey would not have been possible without an impressive group of supporters. This dissertation and my successful defense of this research is attributed to this team of individuals.

Dr. Michele Weigle has guided my very long, arduous path through the Ph.D. program. Her expertise, leadership, and guidance are without compare. I cannot thank her enough for the patience and direction provided throughout my academic career. In the same vein, my co-advisor, Dr. Michael Nelson provided a wealth of knowledge that was instrumental to the completion of this major milestone. I sincerely appreciate the willingness of the Web Science and Digital Libraries (WS-DL) research group to join me in tackling a cybersecurity challenge. We all learned alot. I would also like to thank my dissertation committee members, Dr. Ross Gore and Dr. Farayaneh Poursardar, for their valuable feedback which helped to significantly improve the quality of my research. I also thank the faculty and staff of the ODU Computer Science Department for the professionalism and support they have provided over the course of my academic tenure.

I would also like to thank my family, friends, and colleagues (current and former) for their moral support, continued motivation, and their understanding throughout my doctoral journey. The demands of family and work almost consumed me, but in the end I finished strong.

To all I have mentioned, as well as those I have not – Thank you for being a part of my success story.

TABLE OF CONTENTS

Page

Chapter Page

LIST OF TABLES

Table Page

Table Page

LIST OF FIGURES

Figure Page

Figure Page

Figure Page

Figure Page

Chapter 1

INTRODUCTION

Windows Updates Just Got Serious: You Have 24 Hours To Comply. On July 16, 2020, the Department of Homeland Security's Cybersecurity and Infrastructure Security Agency (DHS CISA) issued an emergency directive instructing "all government agencies to deploy patches or mitigations for a critical bug in Windows DNS Server within the next 24 hours."

— Christopher C. Krebs, *DHS Emergency Directive (ED 20-03)*, July 16, 2020 [95]

Imagine what could happen if someone was able to intercept and read every piece of your mail without detection, your new bank card, your replacement driver's license or passport, letters from your doctor, or application forms containing private information. It is not hard to understand what that person could learn about you, and what damaging things they could do by copying or tampering with your mail. Now imagine that a hacker could do the same on your organization's network, intercepting and manipulating users' emails and network traffic, making services unavailable, or harvesting users' credentials. In effect, they would be able to seize complete control of your Information Technology (IT). With no need for human interaction, a single compromised machine could become a 'super spreader', enabling the attack to permeate throughout an organization's network within minutes of the first exploit. This potential for a widespread 'cyber pandemic' was realized on Tuesday, July 14, 2020 and in the days following the emergency directive [95] issued by the Department of Homeland Security.

Patch Tuesday [166] is an industry term that refers to the second Tuesday of each month when Microsoft regularly releases software patches (e.g., security updates) for its software products. The July 14, 2020 Patch Tuesday security updates rolled out by Microsoft listed one particularly dangerous critical vulnerability. CVE-2020-1350, or SigRed as it had already become known by security practitioners, scored a perfect 10 out of 10 under the Common Vulnerability Scoring System (CVSS) [114], an industry-standard severity rating, for several good reasons. The software patch for this vulnerability, one of 123 released by Microsoft on the same day, is of particular importance to the enterprise as it is wormable, or self-propagating, and is able to jump across vulnerable machines without any user interaction; potentially compromising an entire organization's network of computers in the process.

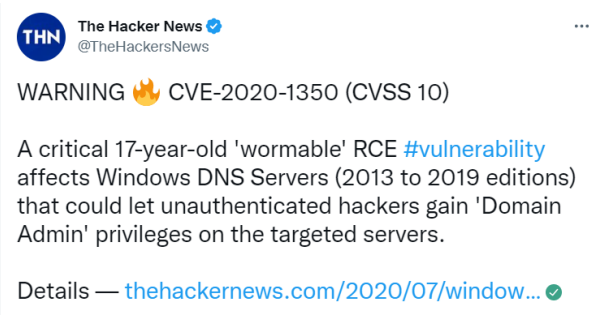
The SigRed vulnerability is considered easy and likely to be exploited. This vulnerability is so likely to be exploited that the U.S. Department of Homeland Security's Cybersecurity and Infrastructure Security Agency (CISA), issued an emergency directive giving government agencies just 24 hours to update their Windows DNS Server software or apply other mitigations. The emergency directive states that the requirements apply to Windows DNS Servers in "any information system, including information systems used or operated by another entity on behalf of an agency, that collects, processes, stores, transmits, disseminates, or otherwise maintains agency information." [95] While this directive itself applies only to relevant U.S. Executive Branch departments and agencies, CISA strongly recommended that state and local governments follow their advice and update as soon as possible. The same guidance was extended to the private sector and individuals running Windows DNS Server. Lamar Bailey, director of security research and development at Tripwire, said, "CVE-2020-1350 is one of the most serious vulnerabilities disclosed this year. It is plausible to believe this is currently being exploited in the wild or will be very soon. It is time to burn the midnight oil and get this patched ASAP." [31] While there was no evidence at the time of release that the vulnerability had been exploited in the wild, the flaw has been hidden in Microsoft's code for 17 years, affecting Windows DNS Server versions 2003 to 2019.

To raise awareness and visibility of CVE-2020-1350 amidst the abundance of software patches released by Microsoft and other software vendors, prominent cybersecurity organizations such as the SANS Institute, CERT-BUND (Computer Emergency Response Team for federal agencies), and The Hacker News (widely-read source of the latest hacking news and cyber attacks) tweeted their own supplemental advisories (Figure 1). While informative, this crowd-sourced approach to vulnerability prioritization is minimally effective because it entails allowing the SANS Institute, DHS, or similar cybersecurity intelligence sources to decide which vulnerabilities are important to a particular organization and worth the level of effort required to remediate.

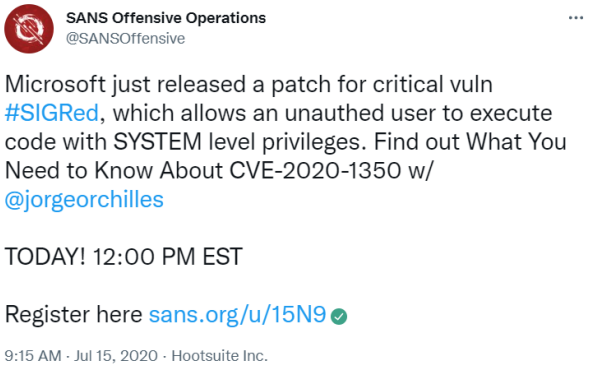
1.1 THE VULNERABILITY PRIORITIZATION PROBLEM

Since the WannaCry [168] and NotPetya [167] malware struck the Internet in 2017, the cybersecurity industry has scrutinized every new Windows bug that could be used to create a similar world-shaking worm. In an article published in *The Atlantic*, security luminary Bruce Schneier asked the question: "Are vulnerabilities in software dense or sparse?" [142] If vulnerabilities are sparse, then every vulnerability disclosed can be fixed to improve the general security of our software. Through patching, organizations can render a vulnerability

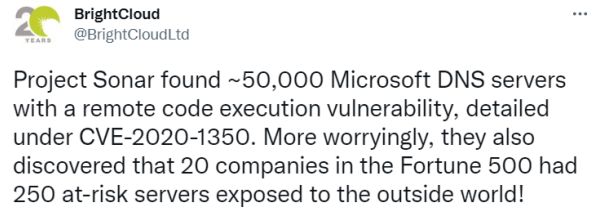
(a) A tweet from CERT-Bund directing followers to Microsoft's portal that provides downloads for security patches (Reproduced from [24]).



(b) A tweet from The Hacker News discussing the vulnerability's exploit potential to use admin privileges (Reproduced from [153]).



(c) A tweet from SANS Pen Test inviting followers to a webinar hosted by a SANS Certified instructor (Reproduced from [137]).



(d) A tweet from BrightCloud discussing the potential impact of the vulnerability and exploit potential among a select group of Fortune 500 companies (Reproduced from [22]).

Figure 1: Tweets alerting organizations to the severity of CVE-2020-1350.

unusable, even if a nation state or cyber criminal already knows about its existence. Many security practitioners know the answer to Schneier's question is that vulnerabilities are dense. They experience the daily deluge of new disclosures. They see it in the backlog of forgotten vulnerabilities found by vulnerability scanners; programs designed to assess computer networks for known weaknesses. In light of such density, frantic efforts to prioritize vulnerabilities may have little to no effect on organizational risk.

1.1.1 PATCH HESITANCY

Vulnerability prioritization is a time consuming process because it is mostly done manually. Since Patch Tuesday updates are delivered in monthly blocks, system administrators often cannot select which patches to apply and which ones are not applicable. System administrators must review the threat posed by each identified vulnerability listed and decide the urgency of patching for their respective organization. Applying the latest patches may take significant time and planning especially if organizations are not on the most recent software versions or have accumulated security patches. Of course, every organization should apply the security updates for their operating systems and critical applications, and they should do so as soon as possible after those updates are released. However, some companies are delaying the automatic installation of security and updates in case the fix proves more troublesome than expected. Numerous instances of Microsoft's Patch Tuesday have led to Recall Thursdays [141] with various patches breaking Microsoft Office [123], affecting the functionality of the Windows operating system, complete system crashes, and the dreaded Blue Screen of Death which occurs when Windows encounters a critical error. Most large enterprises with large IT departments have protocols in place for testing patches before deploying them to the entire network. Smaller businesses, however, do not always have the luxury of sufficient personnel and hardware to set up and maintain a separate testing environment. As a result, many companies have begun to take a "wait and see" approach, delaying patch installation for a week or two in order to let someone else be the "guinea pig." If no major problems emerge in the technology press after that time, they will then proceed to roll out the patches to their own machines.

1.1.2 WHEN EVERY VULNERABILITY IS IMPORTANT

The term zero-day [27] refers to a newly discovered software vulnerability. Because the developer has just learned of the flaw, it also means an official patch or security update to fix the issue has not been released. Therefore, zero-day refers to the fact that developers

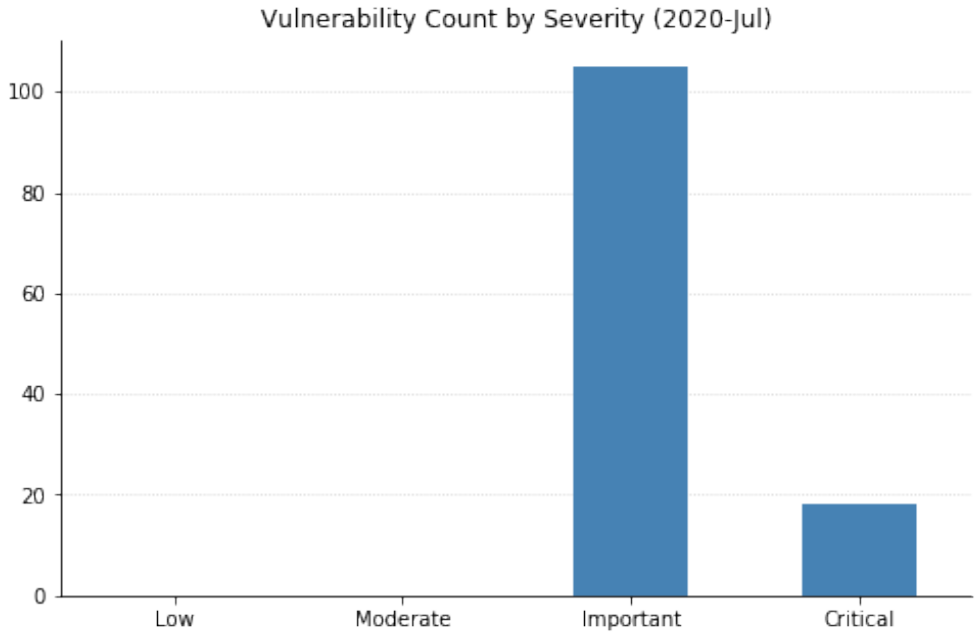
have zero days to fix the problem that has just been exposed and perhaps already exploited by hackers. Once the vulnerability becomes publicly known, the vendor has to work quickly to fix the issue to protect its users. The software vendor may also fail to release a patch before hackers manage to exploit the security hole. Breaches occur because vulnerabilities are exploited. It does not mean attackers used complex or undetectable zero-day exploits to compromise a system and exfiltrate data. It does, however, mean that a weakness existed, an attacker took advantage of it, and the activities went undetected. A successful breach against an organization also does not always mean incompetence or negligence was the root cause. Sometimes organizations with mature technology processes and knowledgeable staff suffer breaches. In recent years, data breaches at places such as at Equifax, Target, and Home Depot [53, 128] have garnered media attention.

Sometimes, there are just too many variables to consider and too much noise regarding the potential impact of a published vulnerability (Figure 2). Of the 123 patches released by Microsoft on Tuesday, July 14, 2020, 18 are listed as Critical and 105 are listed as Important in severity. July 2020 marked five straight months of 110+ CVEs released each month by Microsoft and brought the total for 2020 up to 742. For comparison in Figure 2a, Microsoft released patches for 851 CVEs in all of 2019. With more than 1200 CVEs reported by the end of December 2020 [104], Microsoft also surpassed their totals for 2017 (i.e., 665) and 2018 (i.e., 691). In Figure 2b, the majority of the associated CVEs are rated around 8 on the CVSS scale (0 to 10) which would equate to High impact. The volume of data in the same category can lead to "poverty of attention" [146] that impedes rapid decision making and consumes time and energy in the analysis process.

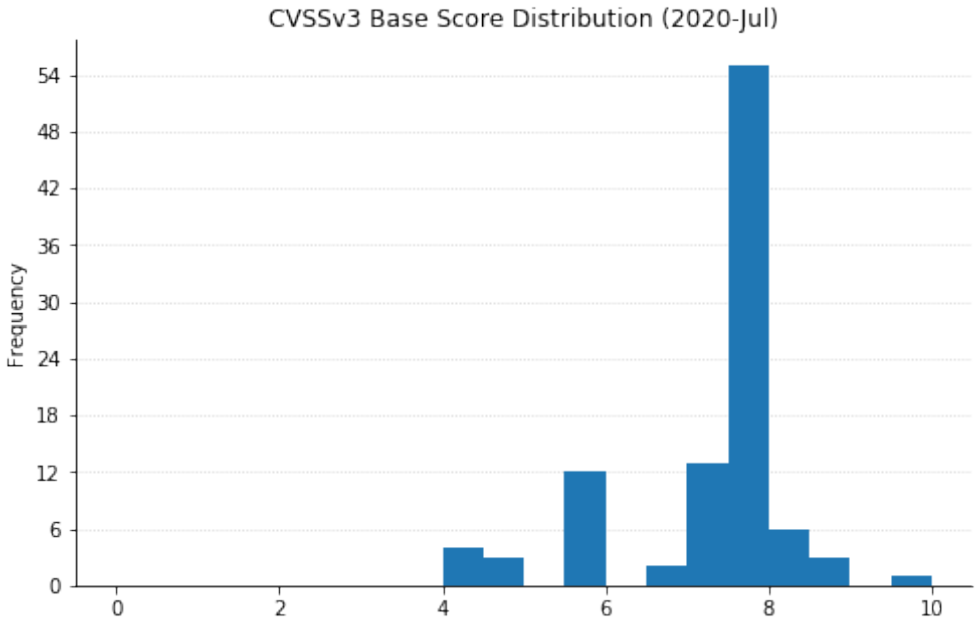
In an information-rich world, the wealth of information means a dearth of something else: a scarcity of whatever it is that information consumes. What information consumes is rather obvious: it consumes the attention of its recipients. Hence a wealth of information creates a poverty of attention and a need to allocate that attention efficiently among the overabundance of information sources that might consume it.

— Herbert A. Simon, Nobel Prize Winner [146]

Successful vulnerability management must balance the two opposing goals of coverage (fix everything that matters) and efficiency (delay or deprioritize what does not matter). These are related to the Information Retrieval (IR) concepts of precision and recall [99]. These opposing objectives are at the crux of the remediation prioritization challenge and the goal of the prediction model we have developed. In driving toward that goal, we have



(a) Severity of released vulnerabilities increasingly rated critical and important.



(b) CVSS scores of released vulnerabilities increasingly rated high and critical.

Figure 2: Review of Microsoft released vulnerabilities in July 14, 2020 Patch Tuesday that includes CVE-2020-1350 (Reproduced from [156]).

examined an array of attributes, assessment techniques, and prioritization strategies. Consideration was given to how quickly vulnerabilities can be exploited, the performance of decisions based on CVSS score thresholds, and differences between vendors.

1.2 CVSS FRAMEWORK AS A RISK MEASUREMENT

The CVSS has been widely used to provide an industry standard way to rank and measure the severity of software vulnerabilities. It is often used to decide which vulnerabilities present the greatest risk and inform patching policies. The CVSS Special Interest Group (SIG) sets the CVSS standard and is composed of representatives from a broad range of industry sectors, from banking and finance to technology and academia.

The CVSS scoring system is now in its third iteration which was released in June 2019. The CVSS Version 3 (CVSSv3) score is calculated by combining characteristics in three groups (base, temporal, environmental) (Figure 3), which are referred to as CVSS metrics using a formula that is publicly available. Analysts use ordinal label assignments and metric scoring groups related to the ease and impact of exploitation. Base metrics represent vulnerability attributes that remain constant over time and user environments (generalized). Temporal metrics represent vulnerability attributes that change with time but not between user environments (e.g., due to changes in publicly available exploit code or a remediation technique). Environmental metrics represent vulnerability attributes that are implementation specific (e.g., the prevalence of a target device within an organization).

A calculated CVSS score ranges in value from 0.0 (least severe) to 10.0 (most severe). The National Vulnerability Database (NVD) performs analysis on CVEs that have been published to the Common Vulnerability Enumeration (CVE) Dictionary. NVD staff are tasked with analysis of CVEs by aggregating data points from the description, references supplied, and any supplemental data that can be found publicly at the time. According to the Forum of Internet Response and Security Team (FIRST), the CVSS is valuable for three main reasons [56]:

- It provides a standardized vulnerability score across the industry, helping critical information flow more effectively between sections within an organization and between organizations.
- The formula for determining the score is public and freely distributed, providing transparency.

- It helps prioritize risk. CVSS rankings provide both a general score and more specific metrics as shown in Figure 3.

In industry, the base CVSS scores have been used directly, and usually without modification to the temporal and environmental metrics, to prioritize vulnerability mitigation strategies. Research has shown that CVSS scores are not strongly linked to the emergence of new cyber exploits and system administrators can be overwhelmed by the volume of vulnerabilities that are nearly indistinguishable based on their high scores [9]. While a CVSS score is indicative of vulnerability severity, it does not predict the exploit potential of the underlying software flaw or the operational impact to the organization. In Figure 4, different scoring vectors can also produce the same CVSS score, which makes it challenging for system administrators to discern, without more in depth analysis, which vulnerability has the greater impact on their operational environment. The lack of clarity is an inherent problem in reducing a multi-dimensional vector to a scalar. While scalars are easy for human discourse, some inherent information is lost. The two vulnerabilities shown in Figure 4 illustrate this problem. CVE-2020-0240 (Figure 4a) refers to a flaw in Google Android devices. A Google Android device could allow a remote malicious user to execute arbitrary code on the system caused by a flaw in the Android framework, a set of system and user interface design tools. By sending a specially-crafted request, an attacker could exploit this vulnerability to remotely execute arbitrary code on the system [122]. CVE-2020-14334 (Figure 4b) refers to a flaw in Redhat Satellite, a systems management tool for Linux-based infrastructure. It allows for provisioning, remote management, and monitoring of multiple Linux deployments with a single centralized tool. The flaw allows a privileged attacker to read cache files to obtain cache credentials that could help the attacker to gain complete control of the Satellite instance. The highest threat from this vulnerability is to data confidentiality and integrity as well as system availability [149]. While both CVE-2020-0240 and CVE-2020-14334 have the same CVSS score, the organization which neither has nor allows Bring Your Own Device (BYOD) connections on their network can safely ignore the Google Android device vulnerability [145].

1.3 IDENTIFYING OPERATIONAL RISK FACTORS

Vulnerability management is the "cyclical practice of identifying, classifying, prioritizing, remediating, and mitigating" software vulnerabilities [55]. Vulnerability management is integral to computer and network security and should not be confused with vulnerability assessment. Risk management is an increasingly important business driver, and stakeholders

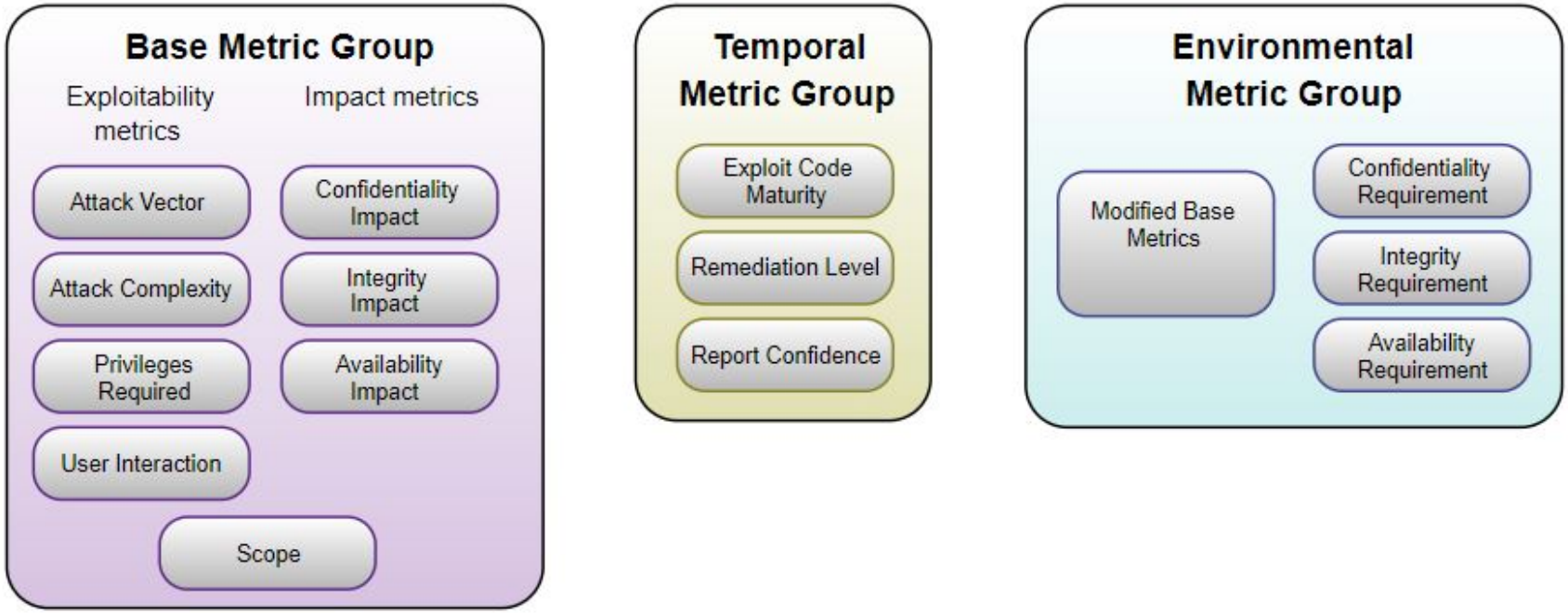
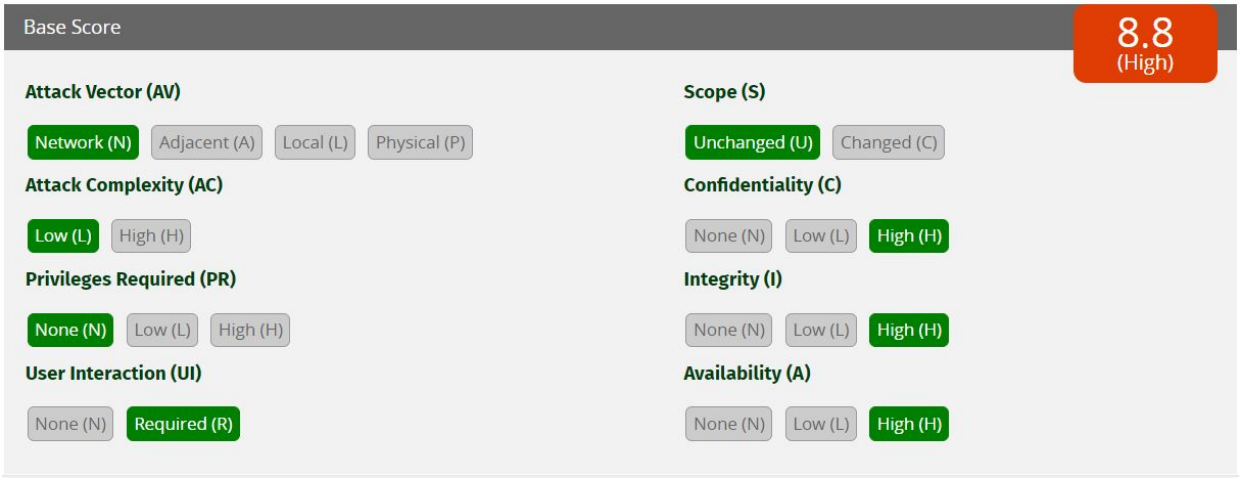
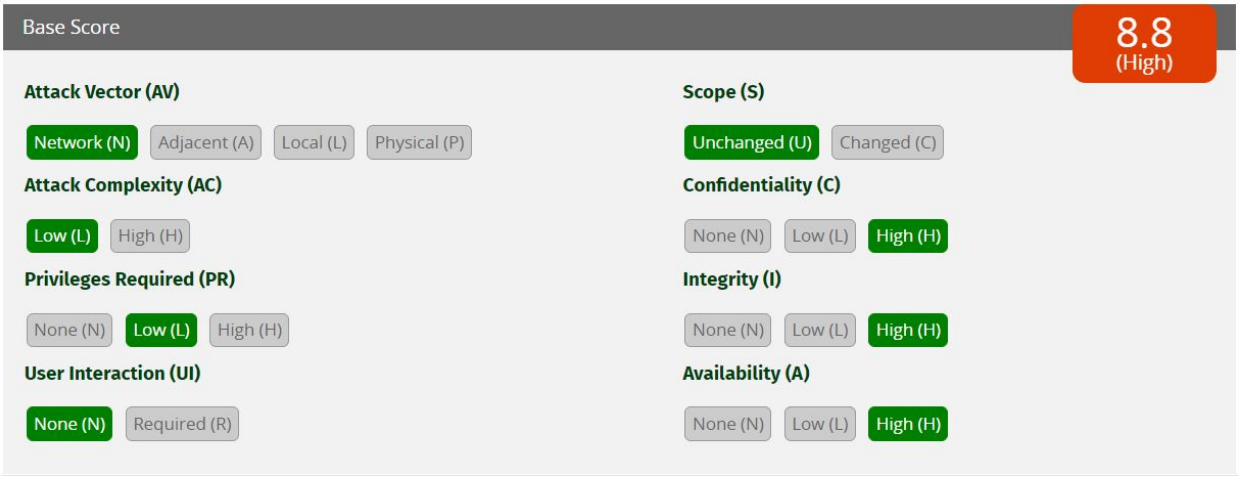


Figure 3: CVSS v3 is composed of three metric groups: Base, Temporal, and Environmental, each consisting of a set of scoring metrics (Reproduced from [56]).



(a) Google Android (CVE-2020-0240) CVSS:3.1/AV:N/AC:L/PR:N/UI:R/S:U/C:H/I:H/A:H [34].



(b) Redhat Satellite (CVE-2020-14334) CVSS:3.1/AV:L/AC:L/PR:L/UI:N/S:C/C:H/I:H/A:H [35].

Figure 4: Vulnerabilities with different CVSS v3 base scoring vectors can produce the same severity score (Reproduced from [34, 35]).

have become much more concerned about risk. Risk may be a driver of strategic decisions, it may be a cause of uncertainty in the organization, or it may simply be embedded in the activities of the organization. An enterprise-wide approach to risk management enables an organization to consider the potential impact of all types of risks on all processes, activities, stakeholders, products and services. The goal of a vulnerability management program is to remediate vulnerabilities in a cost-effective manner before they lead to security incidents. Remediation may involve deploying updates and software patches, modifying system configuration, implementing compensating controls, and a range of other options. The challenge is to identify which vulnerabilities warrant this level of specialized treatment.

An historical view of vulnerability trends can provide insight on past events, however a prediction model must be inherently forward-looking. In other words, we must be able to identify CVEs as likely candidates for exploitation even though they have not yet been targeted by attacks or developed into exploit code. Risk analysis intends to identify and evaluate risks with regards to their impact on the business and further initiate a strategy to mitigate known risks, where necessary. Comprehensive risk assessment of potential risks and further risk management, is a fundamental decision-making process where the organization must thoughtfully consider different types of threats. Examples of cybersecurity threats to organizations can include malicious actions, human error, mechanical failure, and process failure [60]. Conducting risk analysis can be seen as a quantification of uncertainty. In basic terms, the risk is measured by the probability of an event to occur and its consequences. Consequences for an organization's business can consist of different types of losses.

1.4 WANNACRY: THE CASE FOR VULNERABILITY MANAGEMENT

WannaCry was a large-scale malware event that appeared in May 2017, primarily attacking computers running with the Windows 7 or Windows XP operating systems. The malware worked as ransomware. It encrypted the data of the affected computer and asked the user to provide a ransom to decrypt and recover their files, as shown in Figure 5. Researchers will often use unique characteristics discovered in malware or a particular exploit to associate a name (i.e., bug brand). It helps to create an understanding and an ongoing reference point as malware variants surface or activities of a team continue. For traceability, the vulnerability associated with the WannaCry bug brand was assigned CVE-2017-0144.

The timeline, as shown in Figure 6, leading to the exploitation and deactivation of WannaCry was:

Figure 5: Sample screenshot of the WannaCry attack ransom payment procedure (Reproduced from [168]).

- February 10, 2017: Identification of a first version of the ransomware WannaCry on individual computers.
- March 14, 2017: Microsoft responds with security patch CVE-2017-0144 for the currently supported Windows systems (without XP).
- March 27, 2017: A second WannaCry wave without the worm function emerges. Metadata analysis indicates that it was a predecessor of the May campaign and that the same author is behind both campaigns (March and May).
- April 8, 2017: Shadow Brokers release the exploit code for the EternalBlue and DoublePulsar vulnerabilities that enable the worm feature. The exploits were likely stolen from the servers of the National Security Agency (NSA). The NSA is said to have known about the vulnerability for a long time and used it for its own activities [98].
- May 12, 2017: A new version of WannaCry is identified in Palo Alto for the first time. By mid-day, a mass wave of infection occurs in which more than 230,000 computers in over 100 countries are infected.
- May 13, 2017: Microsoft releases security patch for CVE-2017-0144 for Windows XP. However, the spread is mainly prevented by the identification of a Kill Switch URL [171] by the UK researcher Marcus Hutchins. Hutchins discovers the URL in an initial analysis of the malware without knowing its functions. He registers the domain to track malware functionality, unknowingly triggering the Kill Switch.
- May 14, 2017: The author of the malware releases another version of the malware with a modified Kill Switch URL. This URL is also identified and activated on the same day. A short time later, another variation emerges without a Kill Switch called UIWIX [155]. However, this variant was faulty and therefore had no significant impact.
- May 19, 2017: Hackers attempt to neutralize the Kill Switch domain via a variant of the Mirai [172] botnet using a distributed denial-of-service (DDoS) attack. The WannaCry exploit attempt could be prevented by redirecting to a cached version of the page which had a much higher capacity.

WannaCry was able to spread quickly for two reasons:

1. The infected machines had not installed the March 2017 security patch from Microsoft. The operators of the computers had failed or delayed the update in this case. In public

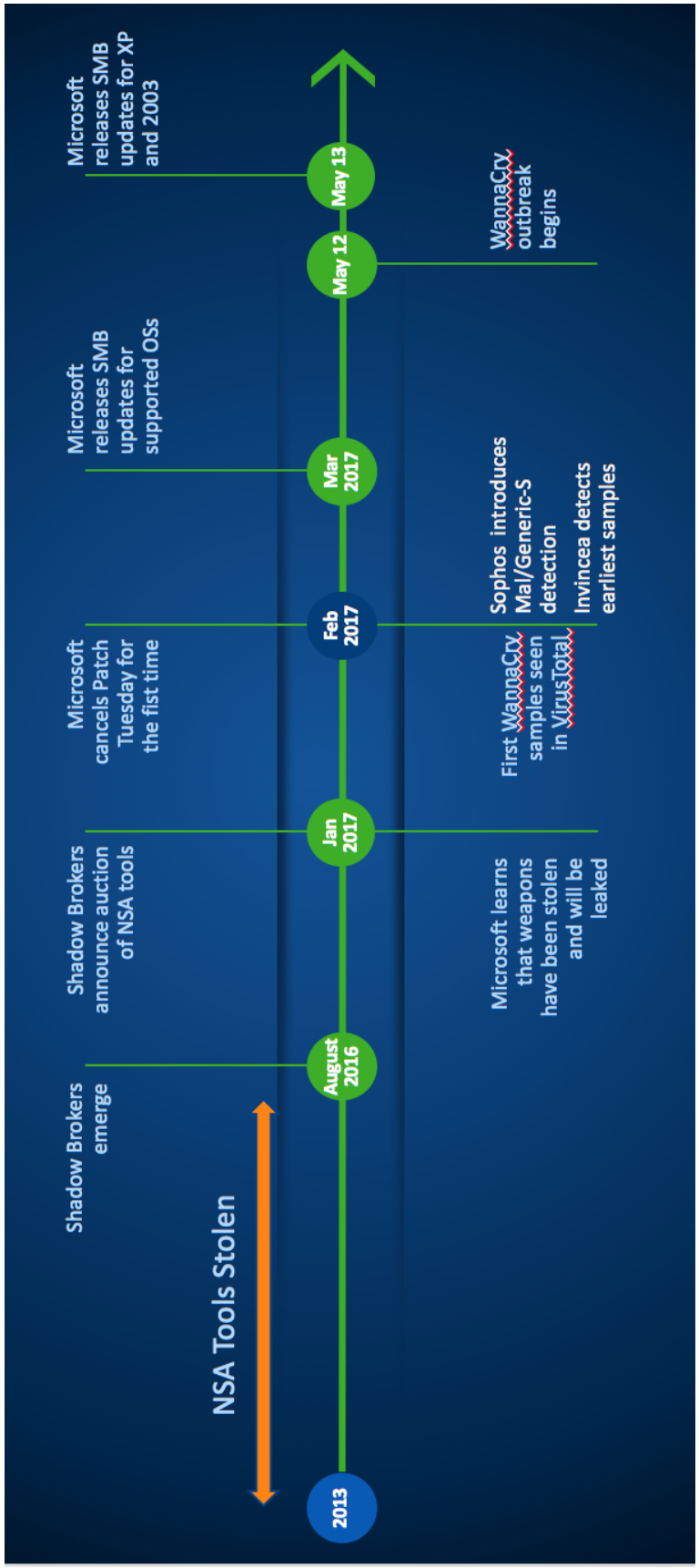


Figure 6: WannaCry timeline (Reproduced from [168]).

authorities and companies in particular, updates are often not implemented at all or only packaged (several updates in one) to avoid possible incompatibilities with existing network configurations and software functions.

2. The common security solutions such as virus scanners, firewalls and intrusion detection systems were not aware of the malware and its functionalities. As a result, WannaCry and its variants were not seen as malware and were subsequently left in the affected systems.

WannaCry provides a real-world example where organizations were breached by known vulnerabilities for which patches existed for months prior to exploitation. Organizations face key challenges when contemplating a remediation strategy that best balances two competing objectives. On one hand, they could attempt to patch all vulnerabilities on the network. While this would provide the greatest coverage of vulnerabilities patched, it would inefficiently consume resources by fixing low-risk vulnerabilities. Conversely, randomly patching a few high-risk vulnerabilities would be highly efficient, but may leave the organization exposed to other high-risk vulnerabilities. Using a large collection of multiple data sets and machine learning techniques, we construct a series of vulnerability remediation strategies and compare how each approach performs as we measure the ranking quality.

1.5 APACHE LOG4J: WHEN VULNERABILITIES HIDE IN PLAIN SIGHT

The reason this is still a problem is that so many companies have not covered the basics, and simply don't understand what's in their software. Our data shows that nearly 40% of Log4Shell downloads are still of vulnerable versions. Meaning there's a high chance that other state and national governments — not just in the U.S. — will be breached in the coming months by bad actors.

— Sonotype CTO Brian Fox [102]

Logging is an important component of the software development process. Application logs reveal information on both internal and external events that are visible to the application during execution. When a bug or anomaly is present in software deployment or a security breach occurs, application logs provide reliable evidence necessary to conduct a thorough root cause analysis of the incident. Apache Log4j is an open-source logging framework used by millions of computers, thousands of websites and applications to log various

data within their application. It is part of the Apache Logging Services, a project of the Apache Software Foundation [14]. While Log4j is maintained by Apache, it is utilized in many vendor applications and appliances as well as in custom-built systems. A vulnerability in such a pervasive and ubiquitous piece of software has the ability to impact companies and organizations, including governments, all over the world [44]. On December 9, 2021, industries worldwide were made aware of a critical vulnerability identified as CVE-2021-44228, a zero-day, remote code execution vulnerability in the Log4j framework. If a cyber-attacker exploits this vulnerability, they can instruct the server that is running Log4j to run any software they want, including malicious software that can completely take over that server. The vulnerability was discovered by researchers from the Alibaba Cloud Security team. It was privately disclosed to Apache foundation on November 24, 2021 and publicly disclosed in the following weeks. The Apache Software Foundation, assigned Log4Shell a CVSS score of 10 (Critical), the highest available score.

This vulnerability, also known as Log4Shell, allows remote code execution in many applications through web requests and without authentication, which enables all the information technology (IT) and operational technology (OT) infrastructure [71]. HTTP requests are frequently logged, and a common attack vector is placing the malicious string in the HTTP request URL or a commonly logged HTTP header, such as User-Agent, X-Remote-IP, or X-Forwarded-For. There are dozens of headers that are typically logged. To exploit the vulnerability, an attacker must cause the application to save a special string of characters in the log. Since applications routinely log a wide range of events such as messages sent and received by users, or the details of system errors, this vulnerability is unusually easy to exploit and can be triggered in a variety of ways. Log4j allows logged messages to contain format strings that reference external information through the Java Naming and Directory Interface (JNDI), Figure 7. This allows naming and directory services to be remotely retrieved across a variety of protocols, including Lightweight Directory Access Protocol (LDAP), Remote Method Invocation (RMI), Domain Name Service (DNS), and Common Object Request Broker Architecture (CORBA). The vulnerability takes advantage of Log4j's failure to validate LDAP and JNDI requests allowing attackers to execute arbitrary Java code on a server. This behavior is a classic example of lapses in input validation and blindly trusting the input without sanitizing the URLs passed in by the input strings.

In the days following the initial announcement that the CVE-2021-44228 exploit was

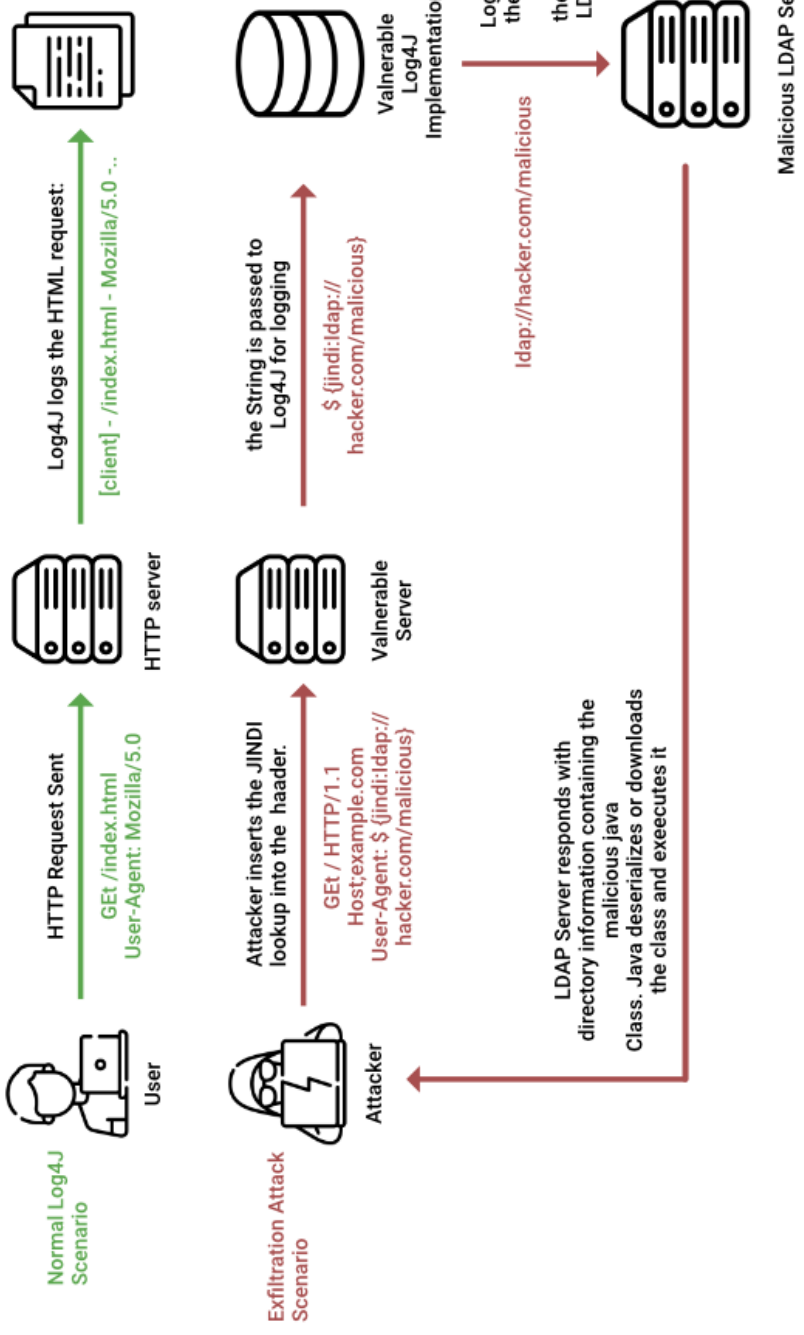


Figure 7: Typical CVE-2021-44228 exploitation attack pattern (Reproduced from [67]).

published, federal employees were asked to quickly help resolve issues that could affect infrastructure and national security. CISA specifically created a web page,[1] *Apache Log4j Vulnerability Guidance*, and established a community-sourced GitHub repository of publicly available information, mitigation strategies, and vendor-supplied advisories regarding the Log4j vulnerability. CISA also issued Emergency Directive (ED) 22-02 requiring federal civilian departments and agencies to assess their internet-facing network assets for the Apache Log4j vulnerabilities and immediately patch these systems or implement other appropriate mitigation measures. The emergency directive was based on several known conditions: (1) the current exploitation of these vulnerabilities (i.e., CVE-2021-44228) by threat actors in external network environments, (2) the likelihood of the vulnerabilities being exploited, (3) the prevalence of the affected software in the federal enterprise, (4) the high potential for a compromise of agency information systems, and (5) the potential impact of a successful compromise [43]. The same sense of urgency was demonstrated by some nation-sponsored actors who saw this new vulnerability as an opportunity to strike before potential targets could identify and patch their affected systems. The potential threat from nation state actors was confirmed via observations from leading cybersecurity firms, Figure 8. Subsequent industry reports linked China's APT41 (i.e., Barium, Wicked Panda/Spider) [143] hacking group and Iranian nation state actor APT35 (i.e., Charming Kitten, Phosphorus) [130] with the breach of at least six United States government networks over a nine month period.

The Log4j vulnerability is part of a broader set of structural issues impacting vulnerability management. The logging service is one of thousands of unheralded but critically important open-source projects that are used across a near-innumerable variety of organizations. These open-source projects are often created and maintained by well-intentioned volunteers and developers, who do not always have adequate resources and personnel for incident response and proactive maintenance even as their projects are critical to the economy. CISA's Cyber Safety Review Board Report on Log4j noted that open source projects generally do not have dedicated coordinated vulnerability disclosure and response teams to investigate root causes of reported vulnerabilities and work to bring them to resolution [44]. Further, vulnerability detection continues to be a challenge for many organizations because while Log4j is used ubiquitously for logging in many applications, the software providers do not always disclose its presence in software notes or bill of materials. As a result, many companies do not actually know if it being used in their systems or not.

[1] `https://www.cisa.gov/uscert/apache-log4j-vulnerability-guidance`

(a) A tweet from CycodeHQ where cybersecurity firm Mandiant states they are seeing nation state actors exploit the Log4j vulnerability (Reproduced from [45]).

(b) A tweet from TheHackerNews warns of attempts by nation-state adversaries and commodity attackers to exploit vulnerabilities in the open source logging framework Log4j (Reproduced from [154]).

Figure 8: Tweets describing exploits of CVE-2021-44228 by nation state actors.

1.6 RESEARCH QUESTIONS

The goal of this research is to demonstrate that aggregating and synthesizing readily accessible, public data sources to provide personalized, automated recommendations that an organization can use to prioritize its vulnerability management strategy will offer significant improvements over what is currently realized using the CVSS base scores. Relevance-based ranking models, like the one we propose to develop in our research, can enable businesses to adopt a proactive strategy for vulnerability management that delivers the most efficient use of their people, tools, time, and ultimately dollars to address the cyber threats that pose the greatest operational risk. In the same way that search engines provide a better ranking of results based on personalization, we propose to improve the ranking of patches by CVSS base score via recommendations that are both "personalized" and "better". Within this context, we seek to define an approach to cybersecurity vulnerability mitigation that improves upon rankings that employ strategies based on the global CVSS metrics associated with known software vulnerabilities published in the NVD. To achieve the primary research goal, we divide the problem into two high-level research questions.

RQ1: What are the factors that can be used to model attack vectors and security threats based on the skill level of a cyber adversary and their motivation to target a specific industry domain (e.g., national defense, higher education, finance, health care)? Each vulnerability creates one or more security threats that introduce a penetration pathway in a network. Therefore, with this objective, we want to develop correlations which define how an adversary capability can exploit a vulnerability to execute a cyber-attack, damage the device, and potentially deny service to the organization or its constituents.

RQ2: What are the characteristics needed to define vulnerability ranking policies that improve the return on investment (ROI) of applied mitigations, compared to traditional CVSS Base Score policies, relevant to the organization's specific mitigation goals and priorities? This research question seeks to define an approach to prioritize the application of software patches to close out the vulnerabilities and prevent potential cyber-attacks. Specifically, we aim to identify which security threat (i.e., vulnerability) is more critical than other identified threats based on the skill and expertise of the adversary.

In Chapter 2, we define the nomenclature that serves as the foundation for this research to be described in later chapters and provides foundational knowledge of the cybersecurity vulnerability management and publicly available threat intelligence sources. Chapter 3 describes the state of the art in vulnerability prioritization methods and current research efforts investigating cyber-domain ontologies, prioritization, and mitigation strategies. Chapter 4 describes preliminary research we performed to address problems in vulnerability prioritization and the initial stages of this research. In Chapter 5, we describe our approach for identifying threats against critical infrastructure sectors and how we will score and evaluate our ranking framework both qualitatively and quantitatively. Finally, in Chapter 6 we define specific ranking profiles based on the installed software for organizations within our test set.

Chapter 2

BACKGROUND

"I would argue that the directive does not go far enough to call out critical vulnerabilities for which proofs of concept may already be published or for which developing an exploit is trivial. Those indeed have a higher chance of being exploited by threat actors in record time."

— Mounir Hahad, head of Juniper Networks' Juniper Threat Labs *SecurityWeek* [94]

In this chapter, we briefly present the necessary terminology and definitions that will be discussed and utilized extensively throughout the preliminary work sections. This study focuses on the vulnerabilities contained in MITRE's Common Vulnerability Enumeration (CVE) List[1] which provides common identifiers for publicly known cybersecurity vulnerabilities. MITRE is a federally funded independent research organization.

2.1 CYBERSECURITY VULNERABILITY EXCHANGE INFORMATION

In order to provide more context to the study and help measure the importance of remediating any specific CVE, we leverage several other authoritative sources for data enrichment. First, we describe the vulnerability data sources shown in Figure 9, their relationships as depicted in Figure 10, and their attributes. The vulnerability data sources include the CVE, CPE, NVD, CWE, and CAPEC.

2.1.1 COMMON VULNERABILITIES AND EXPOSURES (CVE)

The CVE List is widely known and is the authoritative source of publicly known vulnerabilities. We focus our research on discovered and disclosed vulnerabilities contained in the CVE List from MITRE. We do this primarily because CVEs are publicly tracked, readily available, extensive (although not exhaustive), and have become the de facto standard adopted by many other projects and products. MITRE's enumeration [84] (Figure 11), includes a unique identifier (CVE number), a short free-text description, and a list of references to additional details of the vulnerability (in the form of URLs). It should be

[1] `https://cve.mitre.org/cve/`

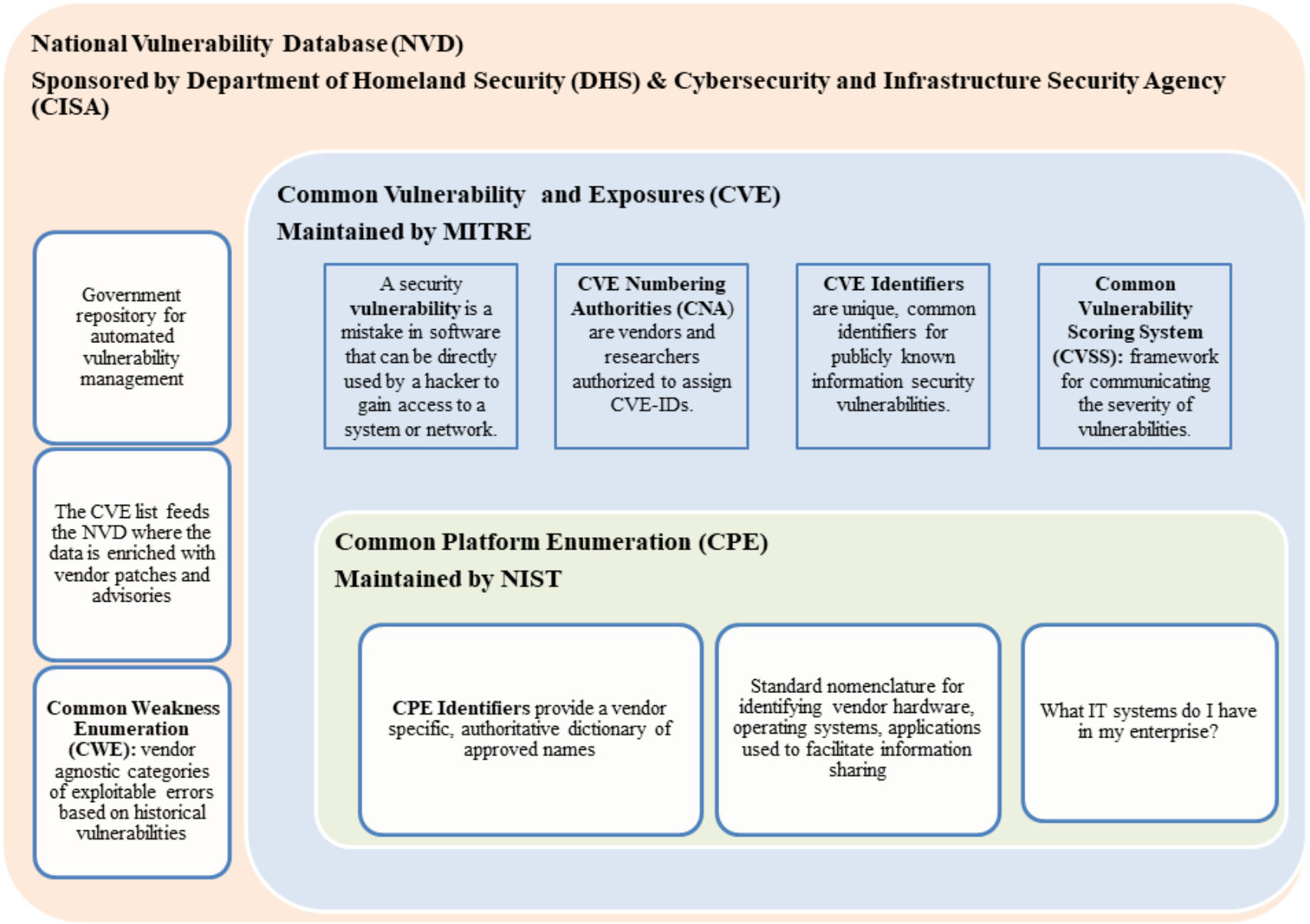


Figure 9: Cybersecurity vulnerability exchange information.

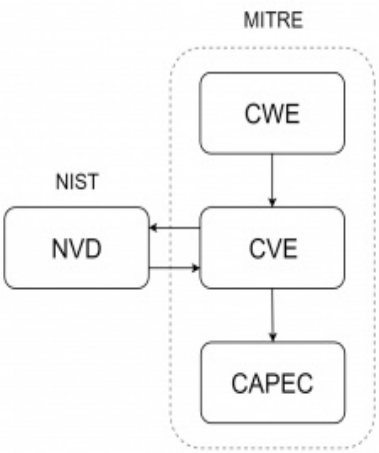


Figure 10: Enumeration database ecosystem (Reproduced from [52]).

noted, however, that CVEs are neither comprehensive nor perfect. Many vulnerabilities are unknown, undisclosed, or otherwise have not been assigned a CVE-ID. Furthermore, CVE listings are curated by humans, which makes them vulnerable to biases, errors, and omissions. Despite these challenges, the CVE List is a valuable community resource that greatly assists the otherwise untenable task of vulnerability management.

The NVD Dashboard [115] shows that through December 31, 2020 over 154,717 CVE entries had been created. Of those, 75,718 have been modified by the submitter and have not yet been re-analyzed by the NVD staff. Another 8,509 have been rejected for various reasons. For all intents and purposes, each of these published CVEs represents a decision and potential action for vulnerability management programs. The criteria for those decisions may be simple in the singular case (e.g., does that vulnerability exist in our environment?), but prove to be quite difficult in the aggregate (e.g., where do we start?). Figure 12 reinforces this challenge by demonstrating the increasing volume of published CVEs over time, another marker for Simon's poverty of attention [146] paradigm.

Figure 12 has two important implications for the volume of published vulnerabilities between 2016 and 2020. First, we have an efficiency problem as marked increases in vulnerabilities to analyze place a significant cognitive load [109] on the security analyst. Remediation must be cost-effective, and Figure 12 illustrates that remediating either comprehensively or randomly would require a great deal of resources. Second, we have a coverage problem. If the focus becomes too narrow or is misplaced, the security program will fail in its mission to remediate the CVE-IDs that are most likely to result in costly security incidents.

Search Results

There are **1** CVE Records that match your search.

Name	Description
CVE-2020-9402	Django 1.11 before 1.11.29, 2.2 before 2.2.11, and 3.0 before 3.0.4 allows SQL Injection if untrusted data is used as a tolerance parameter in GIS functions and aggregates on Oracle. By passing a suitably crafted tolerance to GIS functions and aggregates on Oracle, it was possible to break escaping and inject malicious SQL.

Figure 11: CVE search results using keyword CVE-2020-9402 (Reproduced from [36]).

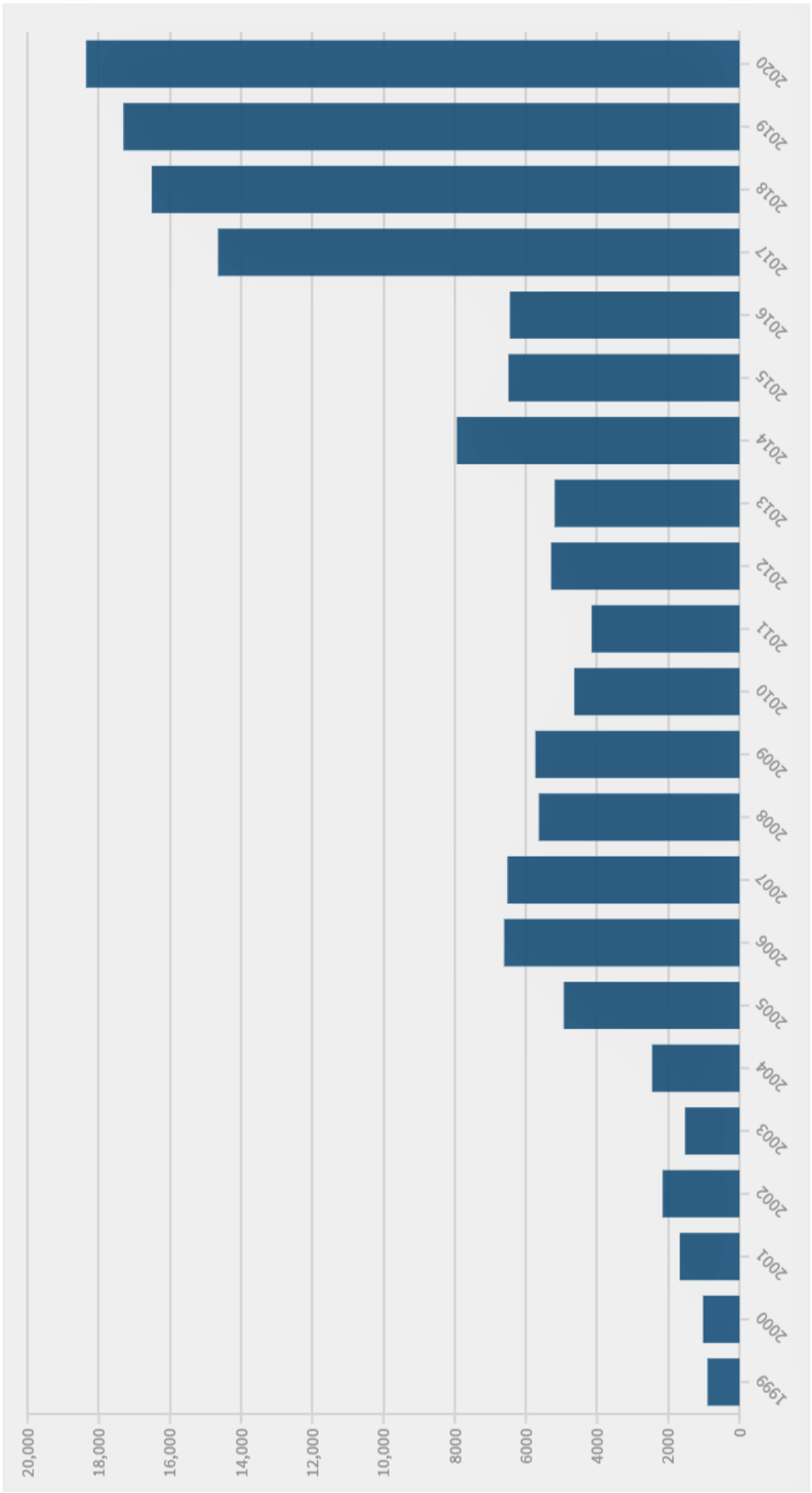


Figure 12: Volume of published CVEs from 1999 through 2020. Note: Vulnerabilities with publish dates before 1999 are not included in this chart (Reproduced from [126]).

Common Vulnerability Scoring System (CVSS)

The CVSS [62, 63, 103, 114, 139], first developed in 2003, is a framework which assesses the severity of information technology (IT) vulnerabilities, giving a severity score to each. The CVSS was developed and is maintained by FIRST [15]. The CVSS has become an international and de facto standard for measuring the severity of a vulnerability. This score is computed using three categories of metrics, which assess the intrinsic characteristics of the vulnerability (base metrics), its change in severity over time (temporal metrics), and the unique user or business environment in which the vulnerability is detected (environmental metrics). CVSS has been designed to assess the impact of atomic attacks on target hosts.

CVSS produces a numeric score between 0.0 (lowest severity) and 10.0 (highest severity) and is fundamentally an ordinal scale, based on six immutable characteristics of a vulnerability, and is independent of any user environmental configurations, security controls, or known exploits. CVSS, the most widespread vulnerability scoring system, is a model for determining the relative likelihood and impact of a given vulnerability being exploited. Among other inputs, the model takes into account impact, complexity, and likelihood of exploitation. Next, it constructs a formula based on these inputs by fitting the metrics, previously discussed in Figure 3, to a desired distribution. This scoring system provides a process to capture the principal characteristics of a vulnerability and produce a numerical score that reflects its severity.

Moreover, when a CVE gets a score, an analyst performs additional research, and assigns a point-in-time likelihood value. The biggest problem with the CVSS model is not the way in which it is executed but rather what it seeks to expose. It is trying to capture (in the temporal component) a snapshot of what the live instances of attacks against these vulnerabilities look like, but it is attempting to do so without looking at any live data. Instead, the CVSS model is a static definition of the very stochastic process of exploit and breach traffic analysis [68].

CVSS 2.0 versus CVSS 3.X

CVSS version 2 was launched in 2007. CVSS version 3.0 was released in June 2015 and was superseded in June 2019 by CVSS version 3.1. The majority of published CVEs were recorded using version 2. However, for our research, we will use only the vulnerabilities published since 2019 scored using version 3.1 of the scoring system. The newer version of CVSS introduces a number of changes in the scoring system that more accurately reflect

vulnerabilities that fall under the web application domain. While all three metric groups, Base Score, Temporal Score, and Environmental Score, remained the same, new metrics such as Scope (S) and User Interaction (UI) were added. In addition, as shown in Figure 13, old metrics such as Authentication (Au) were changed to newer ones such as Privileges Required (PR). Specifically, the following changes were made:

- Confidentiality, Integrity, and Availability metrics were each changed to have scoring parameters of None, Low, or High.
- The Attack Vector metric added the Physical (P) value, which indicates a vulnerability where the adversary must have physical access to a system in order to exploit the vulnerability.
- A new metric, User Interaction (UI), was added. This metric indicates whether or not the cooperation of a legitimate user is needed to conduct an exploit.
- Another new metric, Privileges Required (PR) was added to indicate that administrative or other escalated privileges on the target machine must be achieved in order to successfully exploit the system.
- In the Environmental group, the biggest change was that the environmental metrics in version 2 were completely replaced with a Modified Base Score. Essentially, each of the Base metrics may be modified by the organization to reflect differences between their situation and environment versus others.

The NVD provides qualitative severity rankings of "Low", "Medium", and "High" for CVSSv2 base score ranges in addition to the severity ratings for CVSSv3 as defined in the FIRST Common Vulnerability Scoring System v3.1: Specification Document [56]. With CVSSv3.1 the 0.0 to 10.0 scoring range is now mapped to five different qualitative severity ratings, Table 1.

One observation of CVSSv3 is that the change in scoring methodology increased the severity of many vulnerabilities from high to critical. Cisco conducted a study on this topic and found that the average base score increased from 6.5 in CVSSv2 to 7.4 in CVSSv3 [138]. This means that the average vulnerability increased in qualitative severity from medium to high. The same study concluded that far more vulnerabilities increased in severity than decreased (Figure 14). Anwar et al. [13] similarly noted the small proportion of low severity vulnerabilities in both CVSSv2 and CVSSv3 might suggest some bias against

Base Vector		Temporal Vector	
CVSS V2	**CVSS V3**	**CVSS V2**	**CVSS V3**
Access Vector (AV)	Attack Vector (AV)	Exploitability (E)	Exploit Code Maturity (E)
Access Complexity (AC)	Attack Complexity (AC)	Remediation Level (RL)	Remediation Level (RL)
Authentication (Au)	Privileges Required (PR)	Report Confidence (RC)	Report Confidence (RC)
	User Interaction (UI)	**Environmental Vector**	
Confidentiality Impact (C)	Confidentiality (C)	**CVSS V2**	**CVSS V3**
Integrity Impact (I)	Integrity (I)	Collateral Damage Potential (CDP)	Modified Base Metrics (M*)
Availability Impact (A)	Availability (A)	Target Distribution (TD)	
	Scope (S)	Confidentiality Requirement (CR)	Confidentiality Requirement (CR)
		Integrity Requirement (IR)	Integrity Requirement (IR)
		Availability Requirement (AR)	Availability Requirement (AR)

Figure 13: Base, temporal, and environmental CVSS vectors in V2 and V3 (Reproduced from [51]).

Table 1: CVSSv3 scoring range [56].

Rating	Lower	Upper
None	0.0	0.0
Low	0.1	3.9
Medium	4.0	6.9
High	7.0	8.0
Critical	9.0	10.0

discovering, reporting, or disclosing less urgent security issues. Likewise, the skew towards higher severity ratings in CVSSv3 could induce different vulnerability remediation behavior, as many vulnerabilities rated as medium under CVSSv2 but higher under CVSSv3 might have previously been ignored by security practitioners.

Change	Number of Vulnerabilities That Changed	Percentage (In Comparison to the Total Number of Vulnerabilities)
↑ Medium to high or critical	144	19.33%
↑ Low to medium	35	4.70%
↓ High or critical to medium	12	1.61%
↓ Medium to low	7	0.94%

Figure 14: Change in scoring severity CVSSv2 to v3. Nearly 25% of vulnerabilities increased in severity versus less than 3% that decreased (Reproduced from [17]).

Common Platform Enumeration (CPE)

The Common Platform Enumeration provides a standard machine-readable format for encoding names of technology products, platforms, and vendors. It was developed at MITRE, but ongoing development and maintenance is now handled by NIST. CPE identifies abstract classes of products, such as XYZ Visualizer Enterprise Suite 4.2.3, XYZ Visualizer Enterprise Suite (all versions), or XYZ Visualizer (all variations) [23]. Currently, two versions of the CPE specification are in use: CPE 2.2 and CPE 2.3. The current version 2.3 defines a stack formed by five specifications, including the CPE naming specification [25] and the CPE dictionary specification [29]. Version CPE 2.3 is defined through a set of specifications in a stack-based model where capabilities are based on simpler, more narrowly defined elements that are specified lower in the stack. This design opens opportunities for innovation, as novel capabilities can be defined by combining only the needed elements, and the impacts of change can be better compartmentalized and managed. Figure 15 shows the current CPE 2.3 stack with the most fundamental layer (Naming) at the bottom. Each higher layer (Table 2) builds on top of the layers below it.

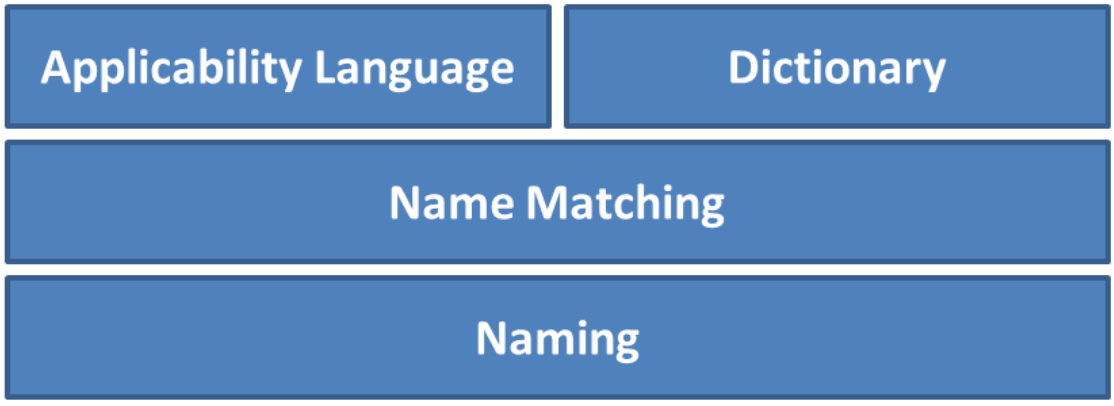


Figure 15: CPE 2.3 stack with the most fundamental layer (Naming) at the bottom. Each higher layer builds on top of the layers below it (Reproduced from [162]).

The WFN format for the software product Adobe Acrobat is shown in Figure 16. The value of the attribute *part* indicates that the IT asset is an application. The logical value

Table 2: CPE 2.3 stack descriptions [136].

Stack	**Description**
Naming	The Naming specification defines the logical structure of Well-formed Names (WFNs), Uniform Resource Identifier (URI) bindings, and formatted string bindings, and the procedures for converting WFNs to and from the bindings.
Name Matching	The Name Matching specification defines the procedures for comparing WFNs to each other so as to determine whether they refer to some or all of the same products.
Dictionary	The Dictionary specification defines the concept of a CPE dictionary, which is a repository of CPE names and metadata, with each name identifying a single class of IT product.
Applicability Language	The Applicability Language specification defines a standardized structure, also known as applicability statements, for forming logical expressions out of WFNs.

NA means not applicable or not used. NA is assigned to attributes that have no meaning for a software product. The logical value ANY indicates that there are no restrictions for an attribute.

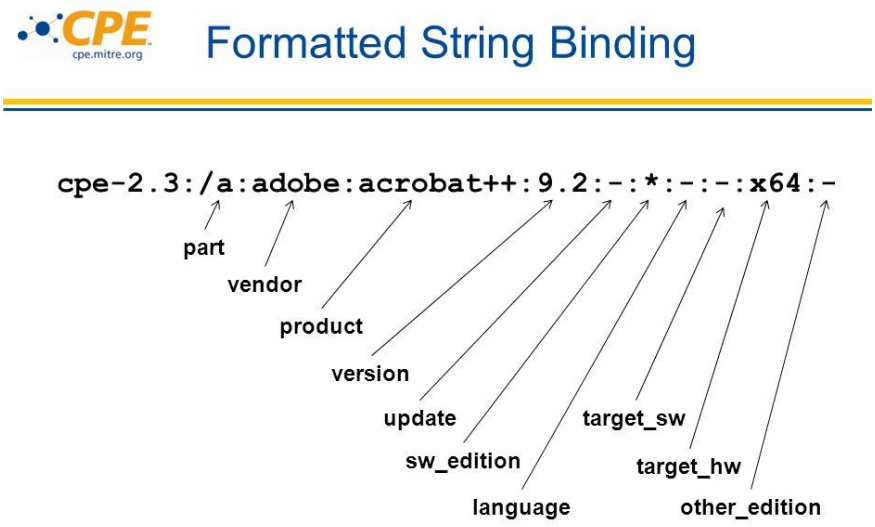


Figure 16: CPE formatted string binding (Reproduced from [97]).

The CPE naming scheme is defined by a set of attributes called Well-Formed CPE Name (WFN). The following attributes comprise this format:

- **Part** — designates an application, operating system, or hardware device
- **Vendor** — identifies the person or organization that manufactured or created the product
- **Product** — identifies the most common and recognizable title or name of the product
- **Version** — vendor-specific alphanumeric strings characterizing the particular release version of the product
- **Update** — vendor-specific alphanumeric strings characterizing the particular update, service pack, or point release of the product
- **Edition** — considered deprecated and should be assigned the logical value ANY
- **Language** — defines the language supported in the user interface of the product being described

- **SW_Edition** — characterizes how the product is tailored to a particular market or class of end users
- **Target_SW** — characterizes the software computing environment within which the product operates
- **Target_HW** — characterizes the instruction set architecture (e.g., x86) on which the product being described or identified by the WFN operates
- **Other** — captures any other general descriptive or identifying information which is vendor- or product-specific and which does not logically fit in any other attribute value

IT management tools can collect information about installed products, identifying these products using their CPE names, and then use this standardized information to help make fully or partially automated decisions regarding the assets. For example, identifying the presence of XYZ Visualizer Enterprise Suite could trigger a vulnerability management tool to check the system for known vulnerabilities in the software, and also trigger a configuration management tool to verify that the software is configured securely in accordance with the organization's policies. This example illustrates how CPE names can be used as a standardized source of information for enforcing and verifying IT management policies across tools.

2.1.2 NATIONAL VULNERABILITY DATABASE (NVD)

The United States NVD is a federal government repository of standards-based vulnerability management data. It is the largest and most comprehensive database of reported known vulnerabilities, both in commercial and open source components. Other open source vulnerability databases aggregate the NVD with additional security advisories (e.g., WhiteSource Vulnerability Database.[2]) The NVD remains a crucial resource in the ongoing struggle to keep applications safe, providing developers and security professionals with the information they need to detect and mitigate newly published known vulnerabilities. This is because the NVD provides an easy to navigate database platform that includes an analysis not found in other public resources. Established in 2005, the NVD is operated under the auspices of NIST. It is sponsored by the Department of Homeland Security's National Cybersecurity and Communications Integration Center, and by Network Security Deployment.

[2] `https://www.whitesourcesoftware.com/vulnerability-database/`

Within a posting on the NVD, security researchers can find a breakdown of many of the details about a software security vulnerability, to help them understand what they are dealing with and what their next steps should be (Figure 17). There are also helpful links to information that is not listed on the NVD itself that lists outside advisories that contain additional solutions and tools. In addition to the basic CVE information provided by MITRE, this research also leverages the details added to each CVE record when published in the NVD. The NVD enriches the base CVE information with details leveraging other community projects which includes scoring and vendor advisories. This additional information includes a description of the CVE and the source of the information, which is generally from the MITRE Corporation. Then, the impact section describes how dangerous a specific vulnerability can be. Based on the CVSSv2 and CVSSv3 Severity and Metrics, the NVD tells readers how the vulnerability has been rated (Critical, High, Medium, Low), as well as details about how the exploitation could actually be carried out.

The NVD is often spoken of interchangeably with the CVE List, however, there are some differences between the two resources despite having a very close relationship. While the NVD is a more robust data set describing the vulnerabilities, the CVE List is more bare bones, providing the straight facts of the CVE-ID number (CVE-year-unique identifier), as well as one public link. Despite their differences, the two databases work hand-in-hand, making the information more accessible for the readers. To put it simply, the CVE dictionary receives submissions and provides IDs for them, while the NVD adds the analysis and makes it easier to search and manage them.

This data enables automation of vulnerability management, security measurement, and compliance. The NVD also integrates the Common Weakness Enumeration (CWE), discussed further in Section 2.2.1, into the scoring of CVE entries, upon which NVD is built, by providing a cross section of the overall CWE structure. NVD analysts score CVEs using CWEs from different levels of the hierarchical structure. This allows analysts to score CVEs at both a fine and coarse granularity, which is necessary due to the varying levels of specificity possessed by different CVEs [114].

When a vulnerability is discovered by a security researcher or company, in many cases they will inform the CVE program to reserve an ID. This information will stay private for a period of 60-90 days to give the owner of the product or open source project time to find a fix to the vulnerability and update relevant vendors if necessary before the word of the exploit becomes public.

It should be said that the NVD will respect the grace period as well, and will delay

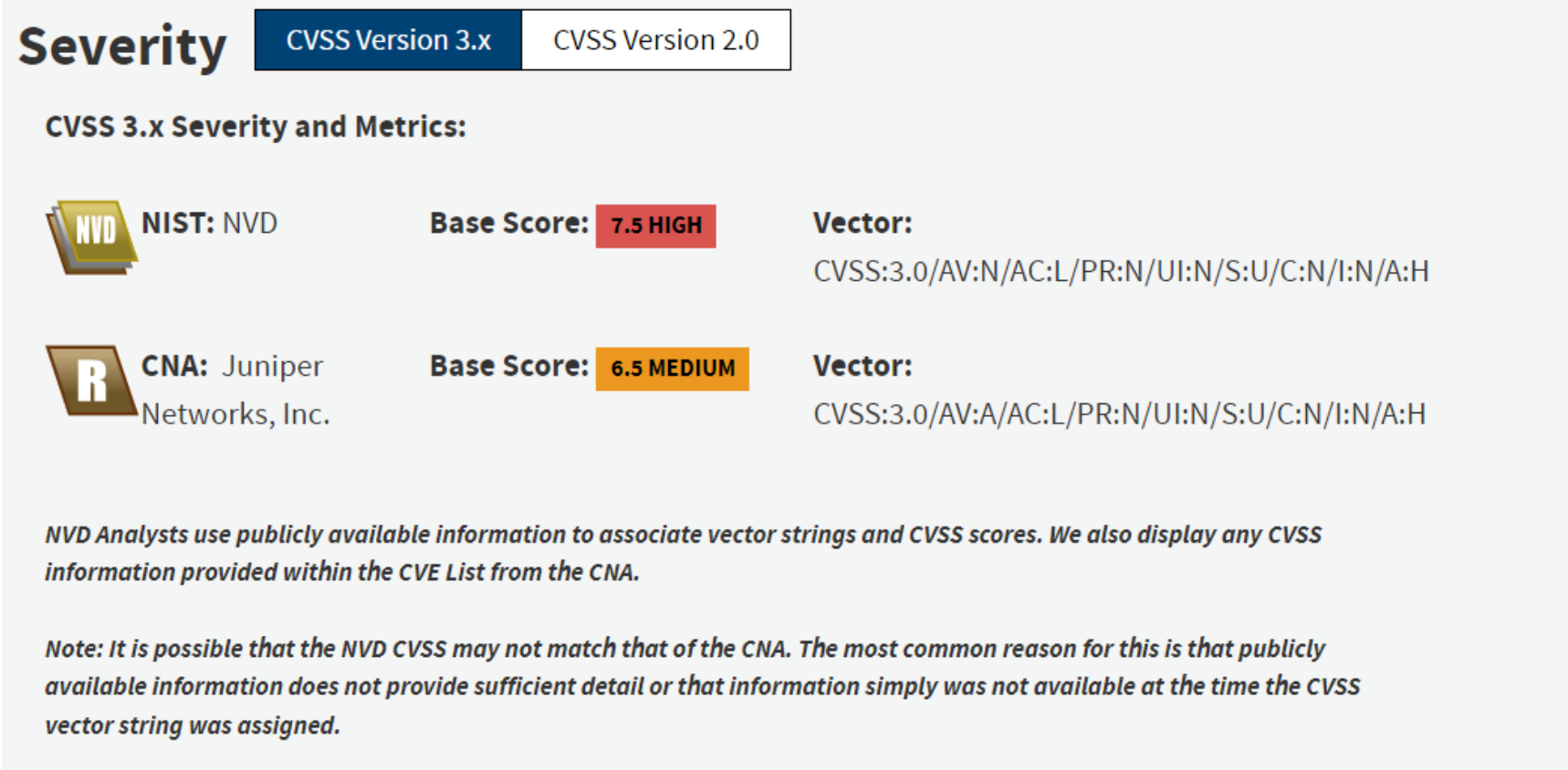


Figure 17: Excerpt of NVD details for CVE-2019-0013 (Reproduced from [33]).

publishing anything until it is no longer "Reserved" by the CVE. Once a CVE entry is posted to the NVD, it will likely stay there unless someone brings a serious dispute to prove that it should be removed. This shows that the NVD has become a fairly exhaustive and dependable database that continues to grow over time. Figure 18 shows the distribution of vulnerabilities by severity during the time period from 2001 to 2020.

2.1.3 SOFTWARE VULNERABILITY LIFECYCLE

Our goal is to remediate vulnerabilities in the most efficient way. But before we discuss prioritization strategies and models, we first need to establish some basics. We start with the meaning of *vulnerability*. The term has wide-ranging usage, including general weaknesses in security posture, various types of code-level flaws, and specific entries in the CVE List and the NVD. A software vulnerability is a programming mistake that allows an adversary access into that system. Heartbleed and WannaCry, discussed in Section 1.4, are recent news worthy examples, however hundreds of vulnerabilities are discovered every year. The CVE defines a vulnerability as "a flaw in a software, firmware, hardware, or service component resulting from a weakness that can be exploited, causing a negative impact to the confidentiality, integrity, or availability of an impacted component or components" [108]. Mitigation of the vulnerabilities in this context typically involves coding changes, but could also include specification changes or even specification deprecations (e.g., removal of affected protocols or functionality in their entirety). All vulnerabilities in the NVD have been assigned a CVE identifier and abide by this more granular definition. In general, a vulnerability will experience some combination of the milestones shown in Figure 19 during its life cycle:

Created — A vulnerability is created when flawed code is written and released in a vulnerable state. It has the potential for exploitation regardless of whether or not it has been discovered, disclosed, or developed into exploit code. MITRE offers the following disclaimer regarding CVE dating: "The entry creation date may reflect when the CVE-ID was allocated or reserved, and does not necessarily indicate when this vulnerability was discovered, shared with the affected vendor, publicly disclosed, or updated in CVE" [135].

Discovery — The means for and statistics surrounding vulnerability discovery are not the focus of this research. When someone learns that a vulnerability exists, it has been discovered.

Disclosure — When a vulnerability is discovered, it may or may not be publicly reported. Vulnerability disclosure is a trending topic in the cybersecurity industry [51, 134], but it is not the focus of our current research. We will only study vulnerabilities that are

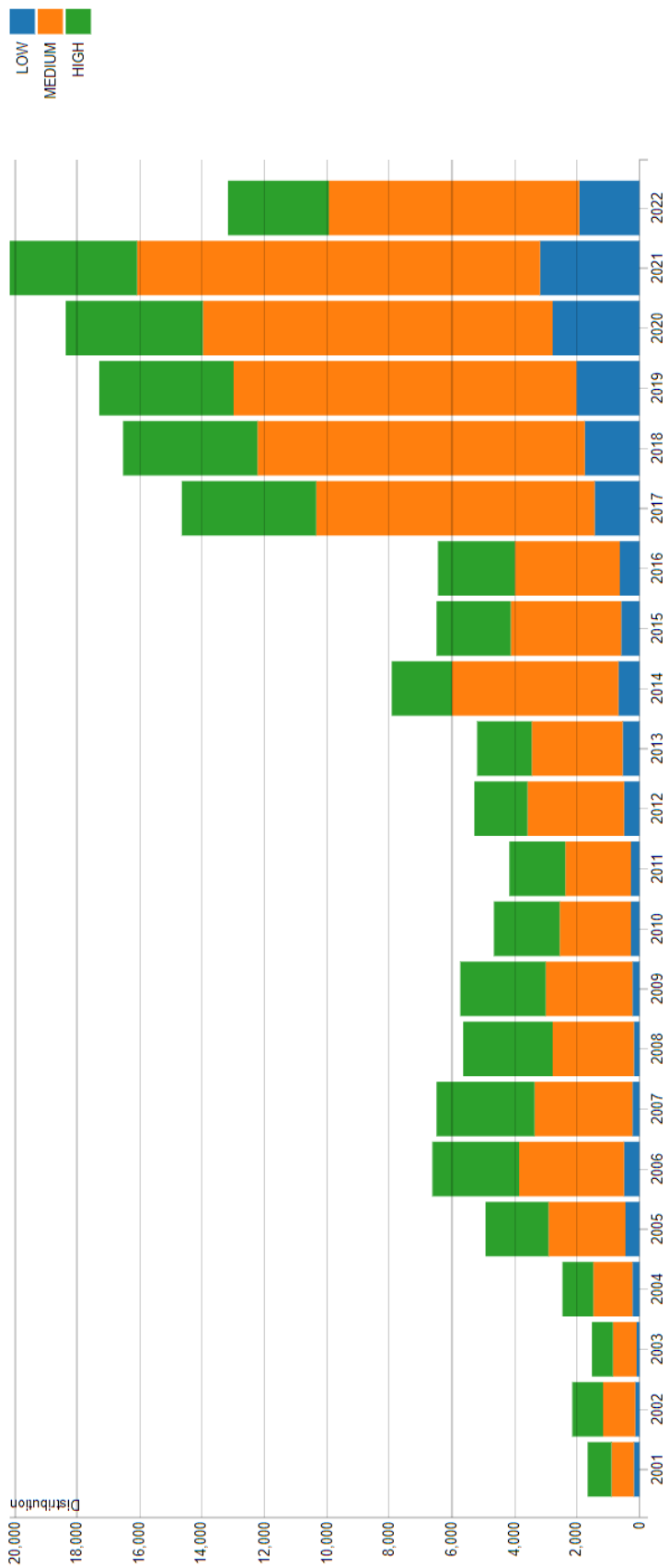


Figure 18: CVSS severity distribution over time. The choice of low, medium, and high is based upon the CVSS V2 base score (Reproduced from [113]).

both discovered and disclosed. For our purposes, disclosure implies that a vulnerability has been reported to CVE Numbering Authorities (CNA) and assigned a CVE-ID. This puts it on the radar for possible remediation. However, it should be acknowledged that other sources of vulnerability disclosures exist. The CNA is an organization responsible for the regular assignment of CVE-IDs to vulnerabilities, and for creating and publishing information about the vulnerability in the associated CVE Record. CNAs are software vendors, open source projects, coordination centers, bug bounty service providers, hosted services, and research groups authorized by the CVE Program to assign CVE-IDs to vulnerabilities and publish CVE Records within their own specific scopes of coverage. Each CNA has a specific scope of responsibility for vulnerability identification and publishing.

Exploit Code — When a vulnerability becomes public, proof-of-concept or working code for exploiting it may be published as well. Exploits can be traded on the dark web [20], shared on above-ground mailing lists, published in blogs, or included in exploitation frameworks. Vulnerabilities reaching this stage are more exploitable, but still may not be used for actual exploitation. Proof-of-concept code, working or weaponized code, and automated exploits incorporated into commodity tools all fall under the common banner of exploitable in our research.

Exploitation — Attacks targeting a vulnerability in the wild constitute exploitation. The success of those attacks is ultimately what vulnerability management programs work to prevent. The good news is that not all organizations are targeted at once. So observing exploits may and should escalate remediation priority. In general, the potential for active exploitation in the wild increases with each of these milestones as exploits become increasingly usable by larger populations of adversaries.

Detection Signature — In order to detect the exploitation of a vulnerability, sensors need to know what to look for. Most tools maintain some form of detection signatures to accomplish this. Unfortunately, these take time to create and distribute, meaning most sensors are not capable of detecting exploitation until a working exploit is published and signatures have been distributed. Of course, some exploits will trigger existing or generic signatures (e.g., SQL injection is rather universal).

Not all vulnerabilities follow all the milestones in the order presented in Figure 19. While the majority of CVEs never proceed beyond disclosure, some proceed straight to active exploitation in the wild. In contrast to the SigRed and WannaCry vulnerabilities mentioned in Chapter 1, CVE-2021-23277 (Figure 20) is one of many critical vulnerabilities for which no known exploits have been recorded [161].

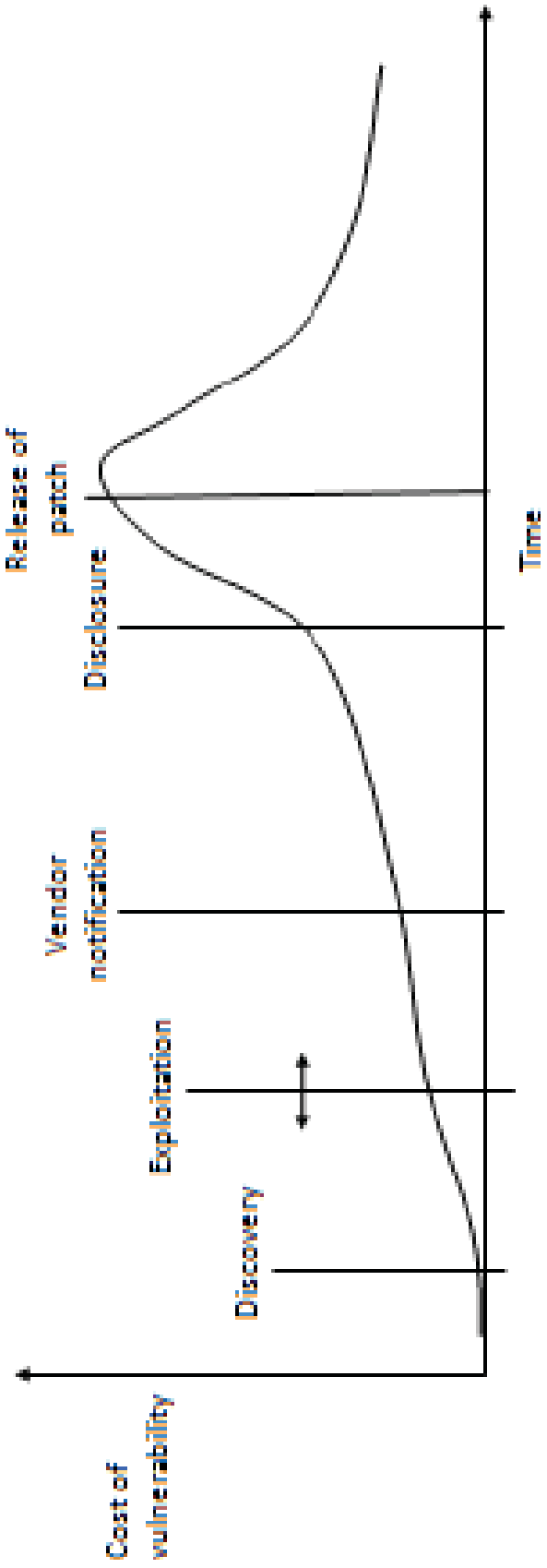


Figure 19: Vulnerability life cycle events (Reproduced from [135]).

CVE-2021-23277 Detail

Current Description

Eaton Intelligent Power Manager (IPM) prior to 1.69 is vulnerable to unauthenticated eval injection vulnerability. The software does not neutralize code syntax from users before using in the dynamic evaluation call in loadUserFile function under scripts/libs/utils.js. Successful exploitation can allow attackers to control the input to the function and execute attacker controlled commands.

+ View Analysis Description

Severity

CVSS Version 3.x | CVSS Version 2.0

CVSS 3.x Severity and Metrics:

NIST: NVD	**Base Score:** 10.0 CRITICAL	**Vector:** CVSS:3.1/AV:N/AC:L/PR:N/UI:N/S:C/C:H/I:H/A:H
CNA: Eaton	**Base Score:** 8.3 HIGH	**Vector:** CVSS:3.1/AV:A/AC:H/PR:N/UI:N/S:C/C:H/I:H/A:H

Figure 20: Critical vulnerability with no known exploits (Reproduced from [37]).

2.2 CYBERSECURITY THREAT INTELLIGENCE INFORMATION

Cyber threat intelligence [30] refers to technology that leverages large-scale threat history data to proactively block and remediate future malicious attacks on a network. Cyber threat intelligence itself is not a solution for vulnerability management. However, because of constantly evolving cyber threats, organizations should avail themselves of massive threat databases that can exponentially improve the efficacy of their own remediation approaches. An ontology must be designed to aggregate and present the data in a comprehensible and usable format. In order to understand how known cyber threats impact the remediation of a CVE-ID, we will centralize the collection of threat data (i.e., weaknesses and attacks) from numerous data sources and formats (Figure 21). The data sources include the CWE, CAPEC, and MITRE ATT&CK.

2.2.1 COMMON WEAKNESS ENUMERATION (CWE)

The Common Weakness Enumeration (CWE) [100] provides a common language for describing software security weaknesses in architecture, design, or code. It was developed and is maintained by MITRE. CWE[3] is an encyclopedia of over 900 types of software weaknesses. Some of the classes are buffer overflow, directory traversal, OS injection, race condition, cross-site scripting, hard-coded password, and insecure random numbers. CWE is a widely-used compilation which has gone through many iterations with the most current list being version 4.3. Each CWE entry (Figure 22) has a variety of information such as description summary, extended description, white box definition, consequences, examples, background details and other notes, recorded occurrences (i.e., CVE-IDs), mitigations, relationships with other CWEs, and references.

The NVD integrates the CWE into the scoring of CVE vulnerabilities by incorporating a cross section of the overall CWE structure which is designated as View ID 1003. This view (i.e., subset of the CWE list) includes 127 of the 916 total CWEs available. A subset of the CWE dictionary is shown below:

- CWE-20: Improper Input Validation[4]
  - The product receives input or data, but it does not validate or incorrectly validates that the input has the properties that are required to process the data safely and correctly.

[3] `https://cwe.mitre.org/`
[4] `https://cwe.mitre.org/data/definitions/20.html`

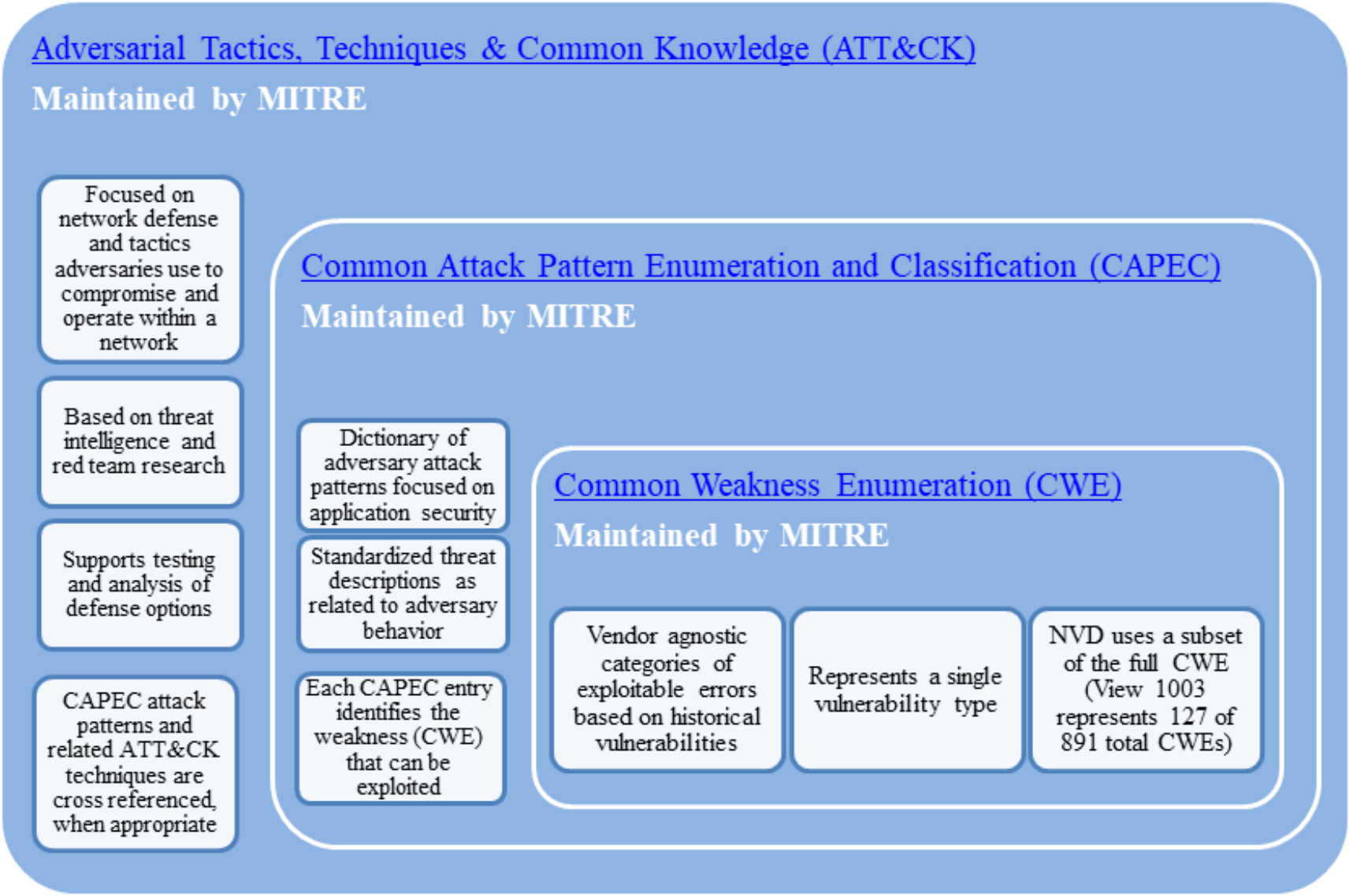


Figure 21: Cybersecurity threat intelligence information.

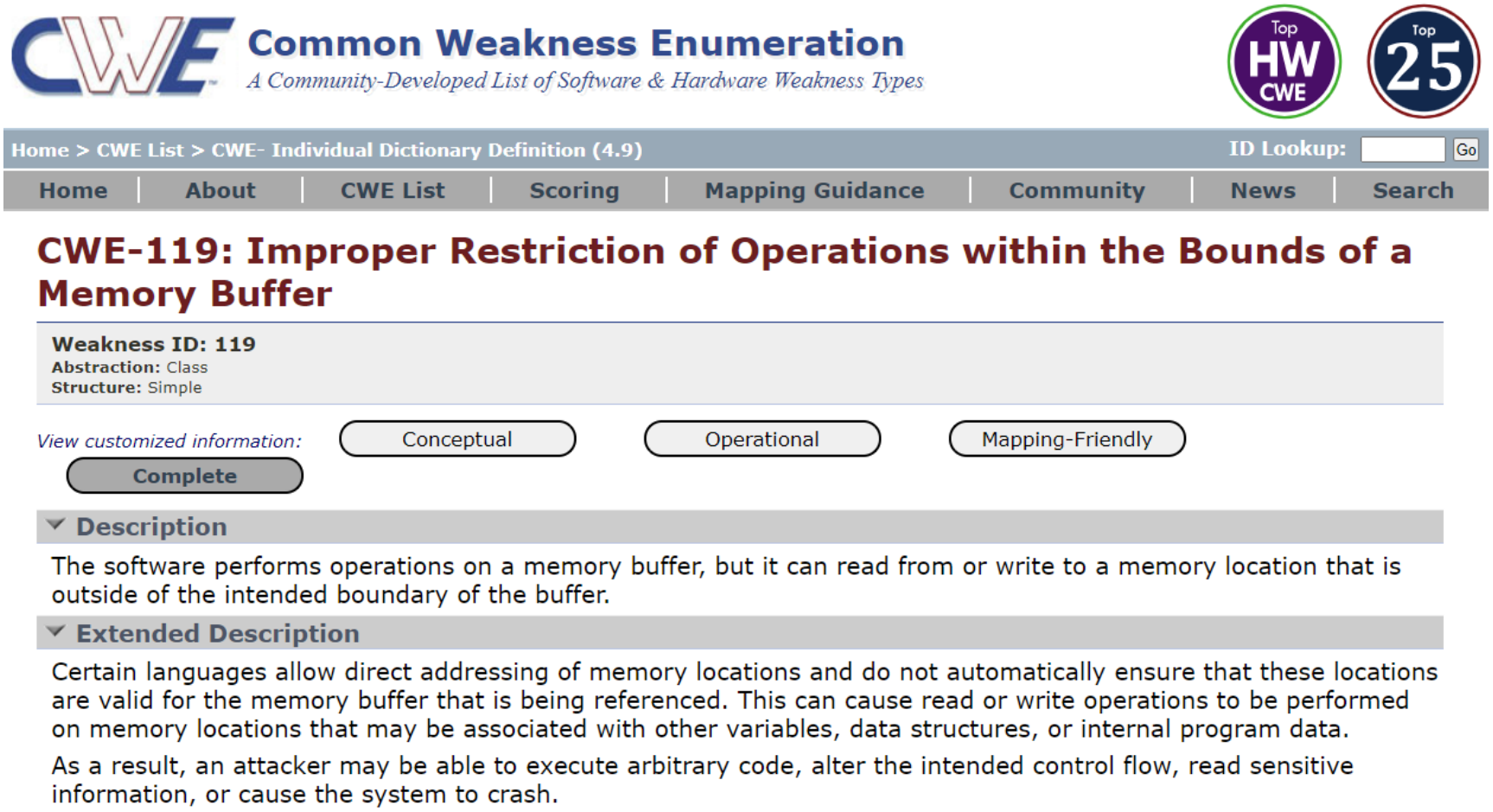


Figure 22: Common weakness enumeration entry (Reproduced from [100]).

- CWE-200: Exposure of Sensitive Information to an Unauthorized Actor[5]
  - The product exposes sensitive information to an actor that is not explicitly authorized to have access to that information.
- CWE-362: Concurrent Execution using Shared Resource with Improper Synchronization ('Race Condition')[6]
  - The program contains a code sequence that can run concurrently with other code, and the code sequence requires temporary, exclusive access to a shared resource, but a timing window exists in which the shared resource can be modified by another code sequence that is operating concurrently.

Figure 23 emphasizes how the assignment of CWEs has changed from year to year. The visualization shows the total number of vulnerabilities assigned a CWE for each year. For example, in 2020 CWE-79 "Cross-Site Scripting" was the identified cause of more than 2000 vulnerabilities. It is possible (although not common) that a vulnerability has multiple CWEs assigned. Understanding the progression of CWE assignments and the associated vulnerabilities may provide insight towards the mitigation strategy intended for this research.

2.2.2 COMMON ATTACK PATTERN ENUMERATION AND CLASSIFICATION (CAPEC)

Attack patterns are descriptions of common methods for exploiting software systems which is becoming increasingly common as malicious individuals and their associated actions are constantly seeking to exploit vulnerabilities in developed software. They derive from the concept of design patterns [66] applied in a destructive rather than constructive context and are generated from in-depth analysis of specific real-world exploit examples. The Common Attack Pattern Enumeration and Classification (CAPEC) [18] "is a comprehensive dictionary and classification taxonomy of known attacks that can be used by analysts, developers, testers, and educators to advance community understanding and enhance defense." CAPEC, sponsored by the United States Department of Homeland Security, is a community-developed list of common attack patterns along with a comprehensive schema and classification taxonomy. Through analysis of observed exploits, the following typical information is captured for each attack pattern [39]:

[5] `https://cwe.mitre.org/data/definitions/200.html`
[6] `https://cwe.mitre.org/data/definitions/362.html`

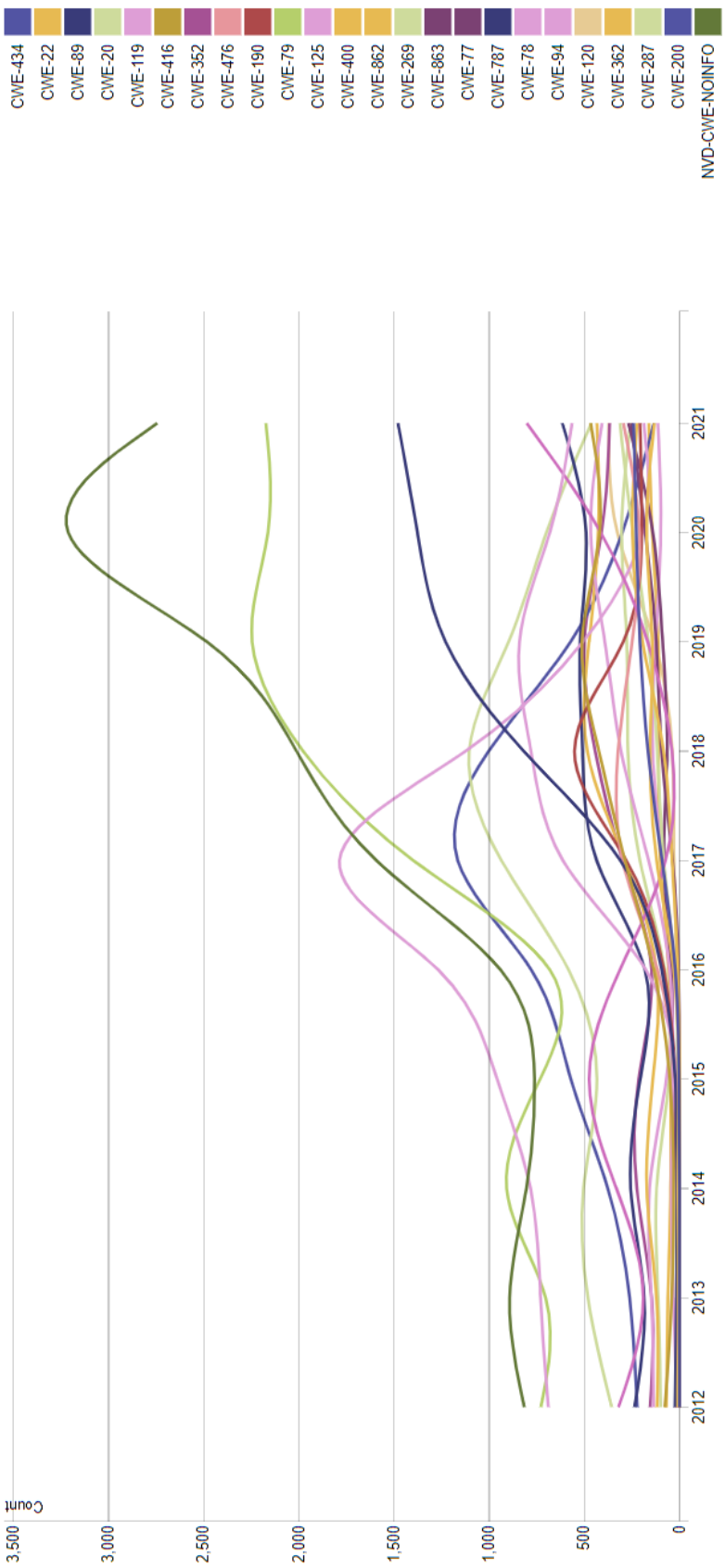


Figure 23: Vulnerability type change by year. The vulnerabilities in the NVD are assigned a CWE based on a slice of the total CWE dictionary (Reproduced from [116]).

- Pattern name and classification
- Attack prerequisites
- Description
- Targeted vulnerabilities or weaknesses
- Method of attack
- Attacker goal
- Attacker skill level required
- Resources required
- Blocking solutions
- Context description
- References

CAPEC uses graph views which are basically hierarchical representations of attack patterns. The top of the hierarchy is a set of categories (Figure 24) under which there are meta-level patterns. These meta-level patterns are parents to standard patterns, which may then be parents to detailed patterns. CAPEC version 3.4 currently provides two views on the CAPEC web site:[7] Mechanisms of Attack and Domains of Attack. In the Mechanisms of Attack view (i.e., View ID: 1000), nine categories are shown at the top level with a total of 527 attack patterns within the entire hierarchy. In the Domains of Attack view (i.e., View ID: 3000), the 527 attack patterns are organized into six categories (Figure 25).

[7] `https://capec.mitre.org/`

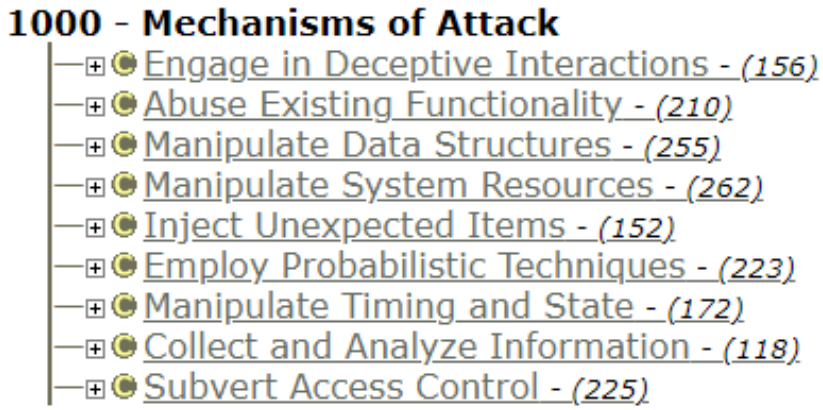


Figure 24: Mechanisms of attack categories (Reproduced from [18]).

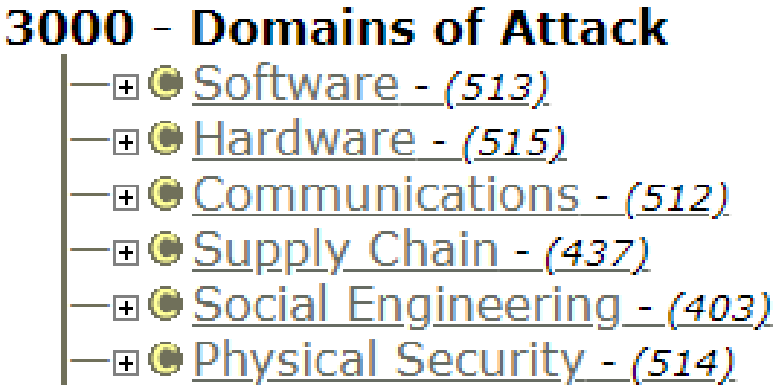


Figure 25: Domains of attack categories (Reproduced from [18]).

Using CAPEC instead of other naming conventions should help analysts better recognize which attack patterns they see most often and then prioritize improvements to their security.

Just knowing there have been a lot of distributed denial-of-service (DDoS) attacks, for example, does not indicate how to best defend against them because this type of incident can occur as a consequence of different attack patterns. For instance, when exploited CVE-2003-0760[8] and CVE-2016-9429[9] can result in a DoS, but each have different attack patterns. The CAPEC attack pattern for CVE-2003-0760 is a UDP Flood[10] (CAPEC-486) while the CAPEC attack pattern for CVE-2016-9429 is Overflow Buffers[11] (CAPEC-100).

2.2.3 MITRE ADVERSARY TACTICS AND TECHNIQUES KNOWLEDGE BASE (ATT&CK)

MITRE ATT&CK[12] is a globally-accessible knowledge base of adversary tactics and techniques based on real-world observations of cybersecurity threats. The ATT&CK knowledge base [148] is used as a foundation for the development of specific threat models and methodologies in the private sector, in government, and in the cybersecurity product and service community. ATT&CK provides a common taxonomy for both offense and defense, and has become a useful conceptual tool across many cyber security disciplines to convey threat intelligence, perform testing through red teaming or adversary emulation, and improve network and system defenses against intrusions. The tactics and techniques are displayed in matrices (Figure 26) that are arranged by attack stages, from initial system access to data theft or machine control.

ATT&CK amasses information that can help analysts understand how attackers behave so they can better protect an organization and defend against cyber threats. ATT&CK's descriptions of tactics, techniques, and procedures (TTPs) provide deep insight into attacker behavior. Tactics describe their goals, like getting inside your network or stealing credentials. Techniques show how they do it. Procedures are highly detailed examples of the tools and actions of specific attacker groups.

[8] `https://nvd.nist.gov/vuln/detail/CVE-2003-0760`
[9] `https://nvd.nist.gov/vuln/detail/CVE-2016-9249`
[10] `https://capec.mitre.org/data/definitions/486.html`
[11] `https://capec.mitre.org/data/definitions/100.html`
[12] `https://attack.mitre.org/`

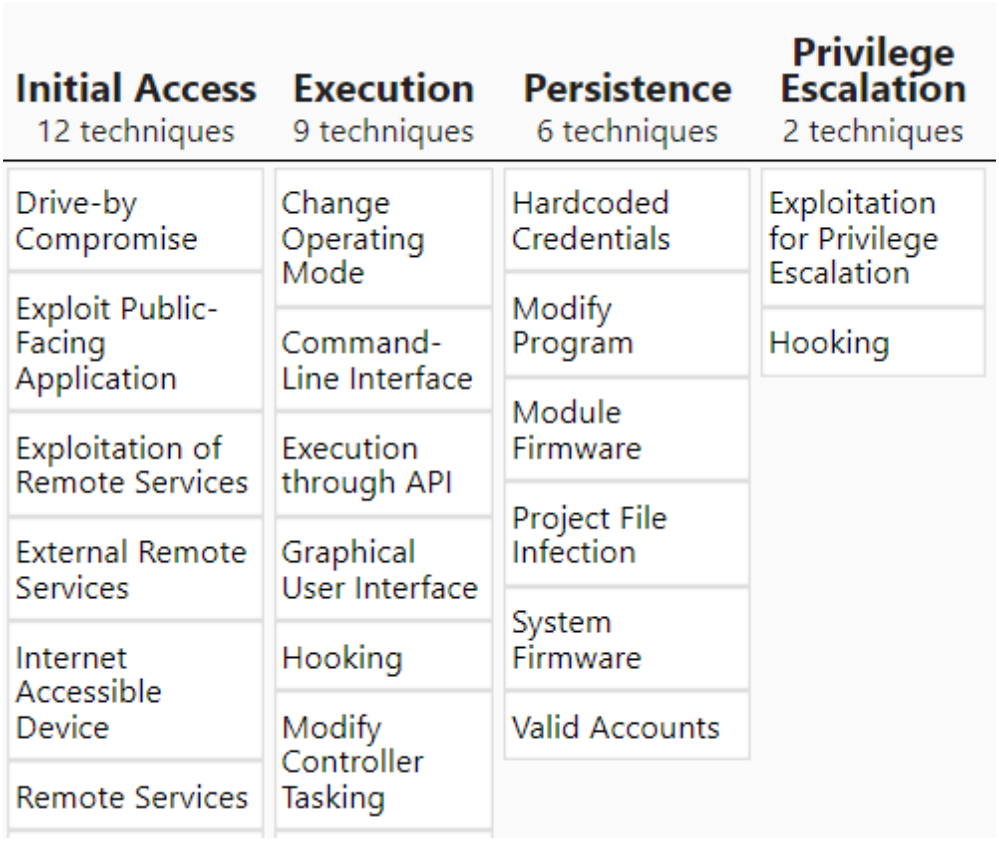


Figure 26: Sample tactics and techniques from the MITRE ATT&CK matrix for enterprise covering techniques against network infrastructure devices (Reproduced from [18]).

Mitigations explain how to defend against attacker TTPs. A single mitigation can apply to multiple TTPs; for instance, multi-factor authentication addresses account manipulation, brute force, external remote services, and many others. At a high-level, ATT&CK is a behavioral model that consists of the following core components:

1. Tactics, denoting short-term, tactical adversary goals during an attack
2. Techniques, describing the means by which adversaries achieve tactical goals
3. Sub-techniques, describing more specific means by which adversaries achieve tactical goals at a lower level than techniques
4. Documented adversary usage of techniques, their procedures, and other metadata

Figure 27 depicts a partial view of the enterprise matrix although not all of the tactics and techniques are visible.

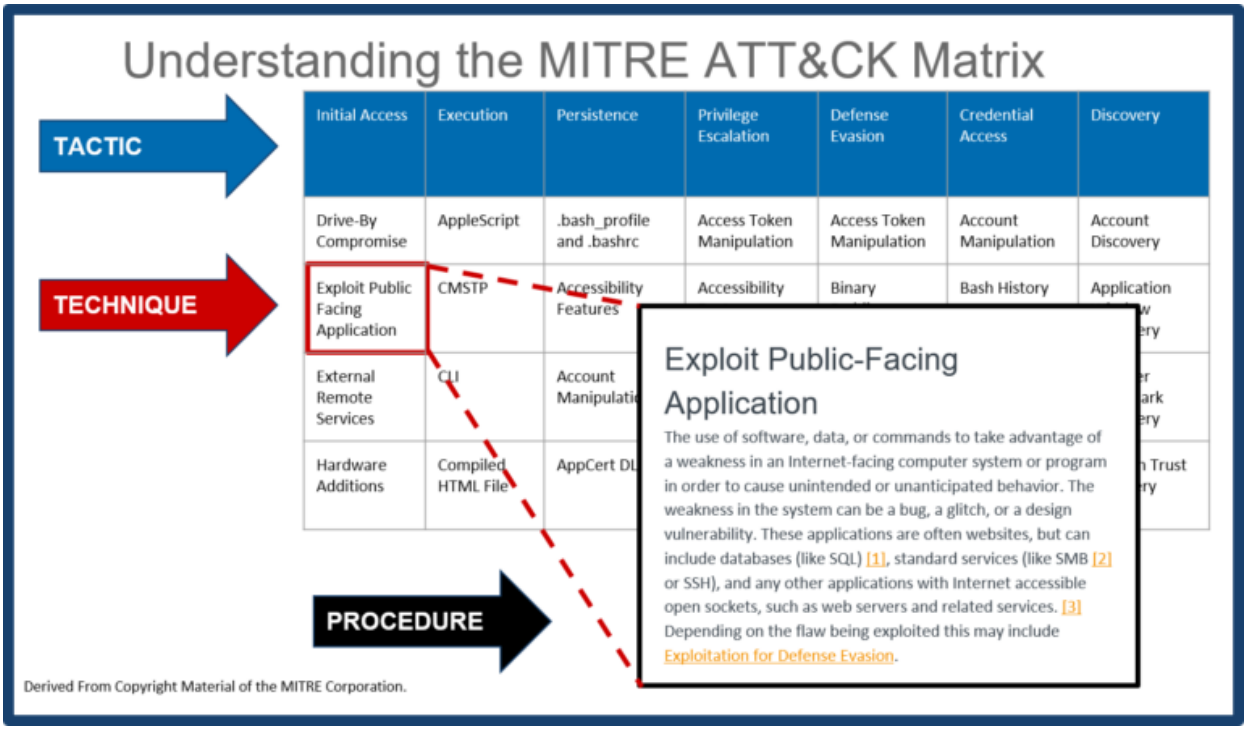


Figure 27: Excerpt from the ATT&CK for enterprise matrix (Reproduced from [79]).

The MITRE ATT&CK Enterprise matrix provides a navigable taxonomy to all techniques that might involve Windows, Mac, and Linux operating systems. Each of the twelve tactics includes between 9 to 67 techniques that might be used. In some cases, techniques may be used by several tactics.

2.3 TRACKING AND PREDICTING EXPLOITS

Basic analysis of CVEs for prioritization may stop at data from MITRE and NVD, but those miss an important part of the equation: what CVEs are attackers actually exploiting in the wild? Unfortunately, no universal source of exploit activity exists, so we have to collect this information through multiple direct and indirect ways. These include host and

network-based detection systems as well as by reverse engineering the malware and tools used by attackers. In this section, we will further discuss how these tools are used for tracking purposes.

2.3.1 EXPLOIT KITS

Exploit kits, such as Angler [77], Blackhole [16], and Neutrino [93], are automated malicious software programs which target client side application vulnerabilities like web browsers, add-ons, Adobe Flash Player, Adobe Reader, or the Java Runtime Environment. Exploit kits are easy to use and do not require in-depth technical knowledge. As a result, they can be used by inexperienced hackers (i.e., script kiddies) as well. Exploit kits mainly take advantage of vulnerable software to gain access to the system.

Sometimes observing exploitation in the wild comes too late for risk-averse vulnerability remediation strategies. In such cases, published exploit code serves as a good indicator of exploitability because it enables attackers to easily weaponize a vulnerability. Sources used to track which CVEs have public exploit code include monthly statistics from the SANS Internet Storm Center [160], Exploit DB [132], and Symantec Attack Signatures [105]. Not only are exploit code releases strongly correlated with active exploitations, but they also indicate the characteristics of a vulnerability that exploit writers target. Tracking the publication of exploit code, therefore, is important to remediation prioritization. Exploit kits can be very sophisticated and they can bypass detection from security products by changing evasion and obfuscation techniques.

The workings of an example exploit kit targeting a website are shown in Figure 28. First attackers exploit server side vulnerabilities and add a malicious hidden iframe in legitimate website. Then, attackers convince users to visit a compromised website. Once the victim visits the site, the page gets redirected to exploit kit hosted on a Bulletproof site. Bulletproof hosting [169] is a service provided by domain and web hosting providers that allows their customers to upload anything, including malicious content. Exploit kits collect various information such as browser version and installed add-ons such as Adobe Flash, Adobe Reader, or JRE version. Based on the collected information, they determine and deliver exploit/malicious code by taking advantage of vulnerable software. If the exploit succeeds, then it can download and install malware, trojans, or spyware.

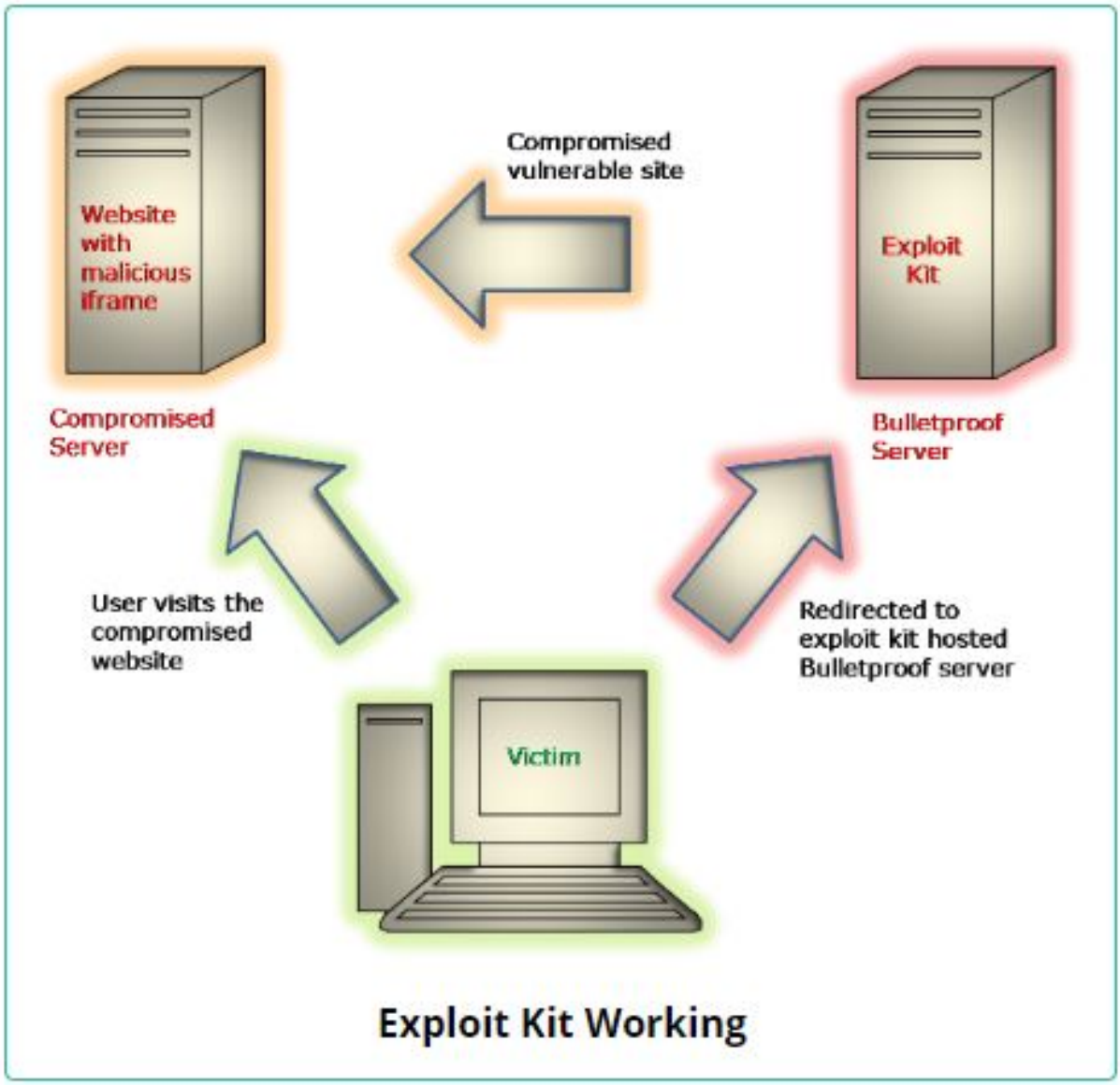


Figure 28: Exploit kit in operation (Reproduced from [69]).

2.3.2 CISA KNOWN EXPLOITED VULNERABILITIES CATALOG

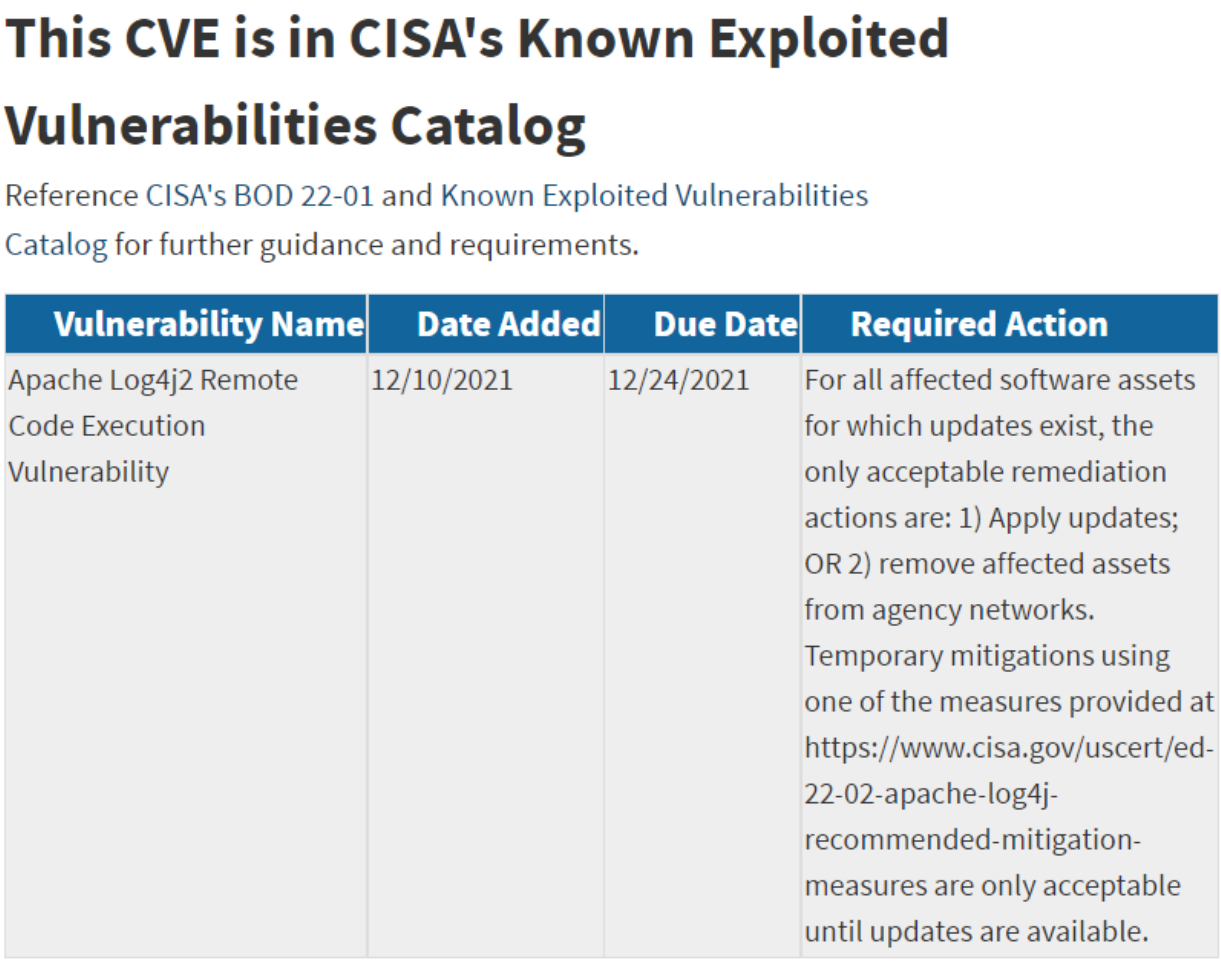
Attackers have used previously published cybersecurity vulnerabilities to compromise systems and networks containing sensitive information. Known vulnerabilities are also potential attack vectors that can provide malicious actors with an opportunity to compromise

critical federal systems and organizational IT infrastructures. On November 3, 2021, the CISA released a new Binding Operational Directive (BOD 22-01) [42]. The directive, titled *Reducing the Significant Risk of Known Exploited Vulnerabilities*, is intended to reduce cybersecurity of already known software flaws and exploited vulnerabilities and is compulsory for all agencies that process federal information through federal information systems. To support BOD 22-01, CISA established a Known Exploited Vulnerabilities (KEV) catalog of relevant vulnerabilities, Figure 29. The catalog serves to focus priority remediation efforts on a subset of vulnerabilities known to be exploited and posing significant risk to the Federal Enterprise. According to the BOD, its mission is to "aggressively remediate known exploited vulnerabilities to protect federal information systems and reduce cyber incidents" [42].

The KEV catalog does not include CVSS scoring because, according to CISA documentation, these rankings do not accurately reflect the frequency with which lower severity vulnerabilities (theoretically, those with less risk) are actually more dangerous due to lack of diligence in applying remediation or exploit chaining [42]. Subsequent to BOD 22-01, the NVD added information to its CVE detail pages to identify vulnerabilities appearing in CISA's KEV Catalog. CVEs appearing in the catalog will now contain a text reference and a hyperlink to the catalog entry, Figure 30.

CVE	Vendor/Project	Product	Vulnerability Name	Date Added to Catalog	Short Description	Action	Due Date
CVE-2021-44228	Apache	Log4j2	Apache Log4j2 Remote Code Execution Vulnerability	2021-12-10	Apache Log4j2 contains a vulnerability where JNDI features do not protect against attacker-controlled JNDI-related endpoints, allowing for remote code execution.	For all affected software assets for which updates exist, the only acceptable remediation actions are: 1) Apply updates; OR 2) remove affected assets from agency networks. Temporary mitigations using one of the measures provided at https://www.cisa.gov/uscert/ed-22-02-apache-log4j-recommended-mitigation-measures are only acceptable until updates are available.	2021-12-24

Figure 29: Known exploited vulnerabilities catalog entry for CVE-2021-44228 (Reproduced from [40]).



This CVE is in CISA's Known Exploited Vulnerabilities Catalog

Reference CISA's BOD 22-01 and Known Exploited Vulnerabilities Catalog for further guidance and requirements.

Vulnerability Name	Date Added	Due Date	Required Action
Apache Log4j2 Remote Code Execution Vulnerability	12/10/2021	12/24/2021	For all affected software assets for which updates exist, the only acceptable remediation actions are: 1) Apply updates; OR 2) remove affected assets from agency networks. Temporary mitigations using one of the measures provided at https://www.cisa.gov/uscert/ed-22-02-apache-log4j-recommended-mitigation-measures are only acceptable until updates are available.

Figure 30: NVD reference to KEV for CVE-2021-44228 (Reproduced from [38]).

2.3.3 EXPLOIT PREDICTION SCORING SYSTEM

The Exploit Prediction Scoring System (EPSS) provides an open, data-driven effort for estimating the likelihood (i.e., probability) that a software vulnerability will be exploited in the wild. This is accomplished by observing and recording exploitation attempts against vulnerabilities and then collecting ancillary information about each of those vulnerabilities. The current EPSS model (v2022.01.01) was trained with 1,164 variables, most of which are boolean values representing the presence of a specific attribute (e.g., was Microsoft the vendor?) [85]. The EPSS analyzes historical events and then makes predictions about future

ones. Multiple data sources are interrogated on daily basis including but not limited to the following list [57]:

- MITRE's CVE List. Only CVEs in the "published" state are scored
- Text-based "Tags" derived from the CVE description and other sources talking about the vulnerability
- Count of how many days the CVE has been published
- Count of how many references are listed in the CVE
- Published Exploit code in any of: Metasploit, ExploitDB, and/or Github
- Security Scanners: Jaeles, Intrigue, Nuclei, sn1per
- CVSS v3 vectors in the base score (not the score or any subscores) as published in the NVD
- CPE (vendor) information as published in NVD
- Ground Truth: Daily observations of exploitation-in-the-wild activity from AlienVault and Fortinet.

Currently, EPSS provides both a probability score, between 0 and 1 (0% and 100%), of observing exploitation activity in the next 30 days, and a percentile which is a rank ordering of probabilities from highest to lowest. Using the published API, the Log4j vulnerability, CVE-2021-44228, would have an EPSS score of 90.4% and percentile of 99.8% on October 6, 2022. The higher the score, the greater the probability that a vulnerability will be exploited within 30 days of publication. Figure 31 depicts the top rated CVEs on a particular date.

Interpreting meaning from probabilities can be challenging when most values are expressed in relatively small range. Percentiles in the EPSS are a direct transformation from probabilities and provide a measure of probability relative to all other scores. The percentile is the proportion of all values less than or equal to the current rank [85]. Figure 32 plots EPSS percentiles (y-axis) versus EPSS probabilities (x-axis).

Top rated CVEs from the last thirty days

We selected the 48 highest rated CVEs published in the last 30 days. They are shown here with the CVE and EPSS score.

CVE-2021-43258 58.1%	CVE-2022-4135 7.9%	CVE-2022-3361 4.6%	CVE-2022-45063 3.8%	CVE-2022-41413 2.6%	CVE-2022-43303 2.5%
CVE-2022-22984 18.3%	CVE-2022-39377 6.7%	CVE-2022-3383 4.6%	CVE-2022-38767 3.6%	CVE-2022-45472 2.6%	CVE-2022-43304 2.5%
CVE-2022-38813 14.5%	CVE-2022-41446 6.7%	CVE-2022-3384 4.6%	CVE-2022-3861 3.5%	CVE-2022-36432 2.6%	CVE-2022-43305 2.5%
CVE-2022-44384 12.7%	CVE-2022-24999 5.7%	CVE-2022-41106 4.5%	CVE-2022-43144 3.1%	CVE-2022-37018 2.6%	CVE-2022-44048 2.5%
CVE-2022-40127 11.8%	CVE-2022-40797 5.6%	CVE-2022-41445 3.9%	CVE-2022-39343 2.8%	CVE-2022-37904 2.6%	CVE-2022-44049 2.5%
CVE-2022-41049 9.1%	CVE-2022-41875 5.6%	CVE-2022-42097 3.9%	CVE-2022-37897 2.7%	CVE-2022-37905 2.6%	CVE-2022-44050 2.5%
CVE-2022-41412 9.0%	CVE-2022-40687 5.2%	CVE-2022-42096 3.9%	CVE-2022-43333 2.7%	CVE-2021-3661 2.6%	CVE-2022-44051 2.5%
CVE-2022-43138 8.4%	CVE-2022-44789 4.8%	CVE-2022-42094 3.9%	CVE-2021-3942 2.7%	CVE-2022-41924 2.5%	CVE-2022-44052 2.5%

Figure 31: Top rated CVEs from the last 30 days as reported on December 4, 2022 (Reproduced from [57]).

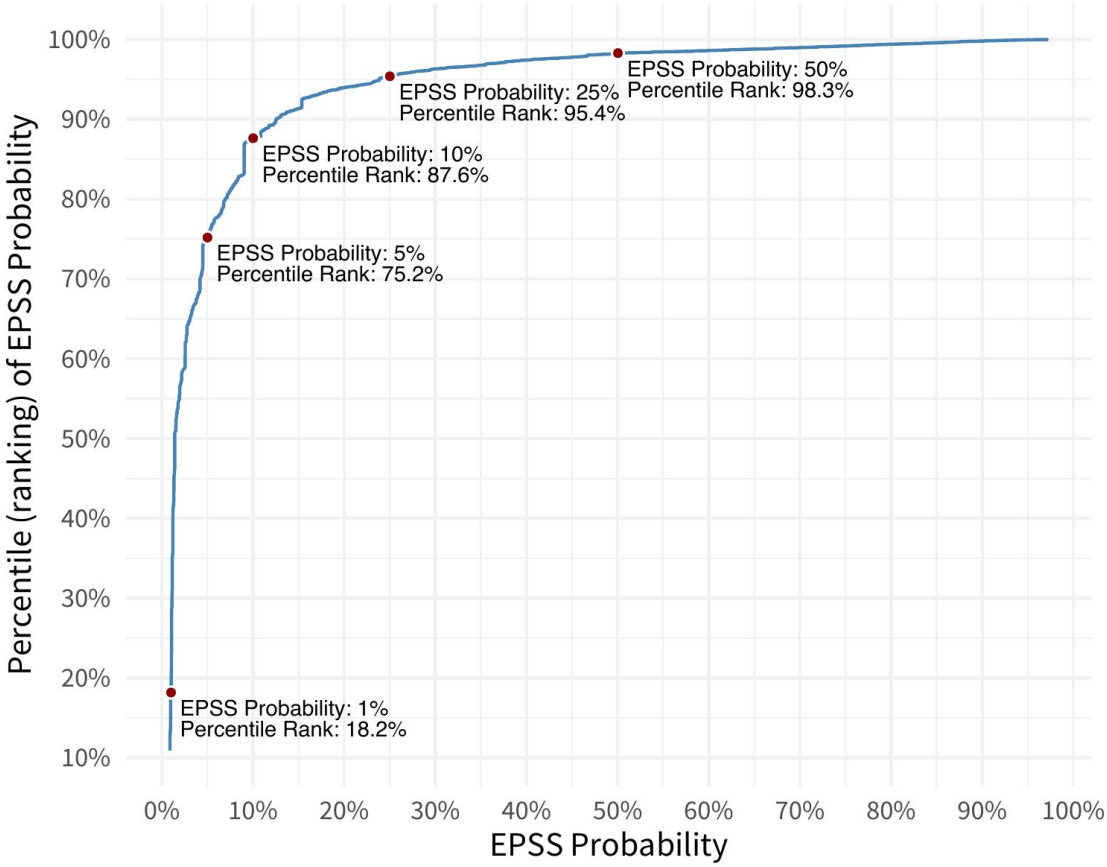


Figure 32: EPSS percentiles versus probabilities (Reproduced from [58]).

An EPSS probability of 10% rests near the 88th percentile which means that 88% of all CVEs are scored lower. A vulnerability with an EPSS probability of 10% sits in the top 12% of all scored vulnerabilities. While 10% is a relatively low probability, for the entire EPSS distribution (170,000+ vulnerabilities), it is among the highest scored [58].

2.4 VULNERABILITY MANAGEMENT

Each year thousands of new software vulnerabilities are reported (Figure 33) requiring organizations to patch operating systems and applications and reconfigure security settings throughout the entirety of their network environment. Actionable threat intelligence needs threat history data for trend analysis. As shown in Figure 33, the volume of threat intelligence data can be overwhelming. All of the CVE-IDs cited in Figure 33 are assigned

by CNAs [5]. To proactively address vulnerabilities before they are utilized for a cyber attack, organizations may perform vulnerability management to achieve the highest levels of security posture possible. Vulnerability management is generally defined as the process of identifying, categorizing, prioritizing, and resolving vulnerabilities in operating systems, enterprise applications, web browsers, and other end-user applications. An ongoing process, vulnerability management seeks to continually identify vulnerabilities that can be remediated through patching and configuration of security settings. A remediation strategy implemented alongside other security tactics is vital for organizations to prioritize possible threats and minimizing their attack surface.

While people working in the many different forms of risk management always have the same goal, to provide a sound basis for decisions on whether risks are acceptable and, if necessary, obtain reliable information concerning how they can be dealt with, there are many different definitions of risk and of the risk management process. For these reasons, the International Standards Organization set out to achieve consistency and reliability in risk management by creating a standard vocabulary that would be applicable to all forms of risk. The definition presented in ISO Guide 73 [82] states that risk is the "effect of uncertainty on objectives". In order to assist with the application of this definition, ISO Guide 73 also states the effect may be positive, negative, or a deviation from the expected, and that risk is often described by an event, a change in circumstances or a consequence. This definition specifically links risks to objectives that are easily applied when the objectives of the organization are comprehensive and fully stated. Even when fully stated, the objectives themselves need to be challenged and the underlying assumptions on which they are based should be vetted as part of the risk management process.

Every new vulnerability introduces risk to the organization. Therefore, a defined process is often used to provide organizations with a way to identify and address vulnerabilities quickly and continually. At a high level, six processes comprise vulnerability management, each with their own sub-processes and associated tasks (Figure 34). The significance of risks increases as vulnerabilities trigger the creation of the associated exploit kits and decreases when vendor patches become available.

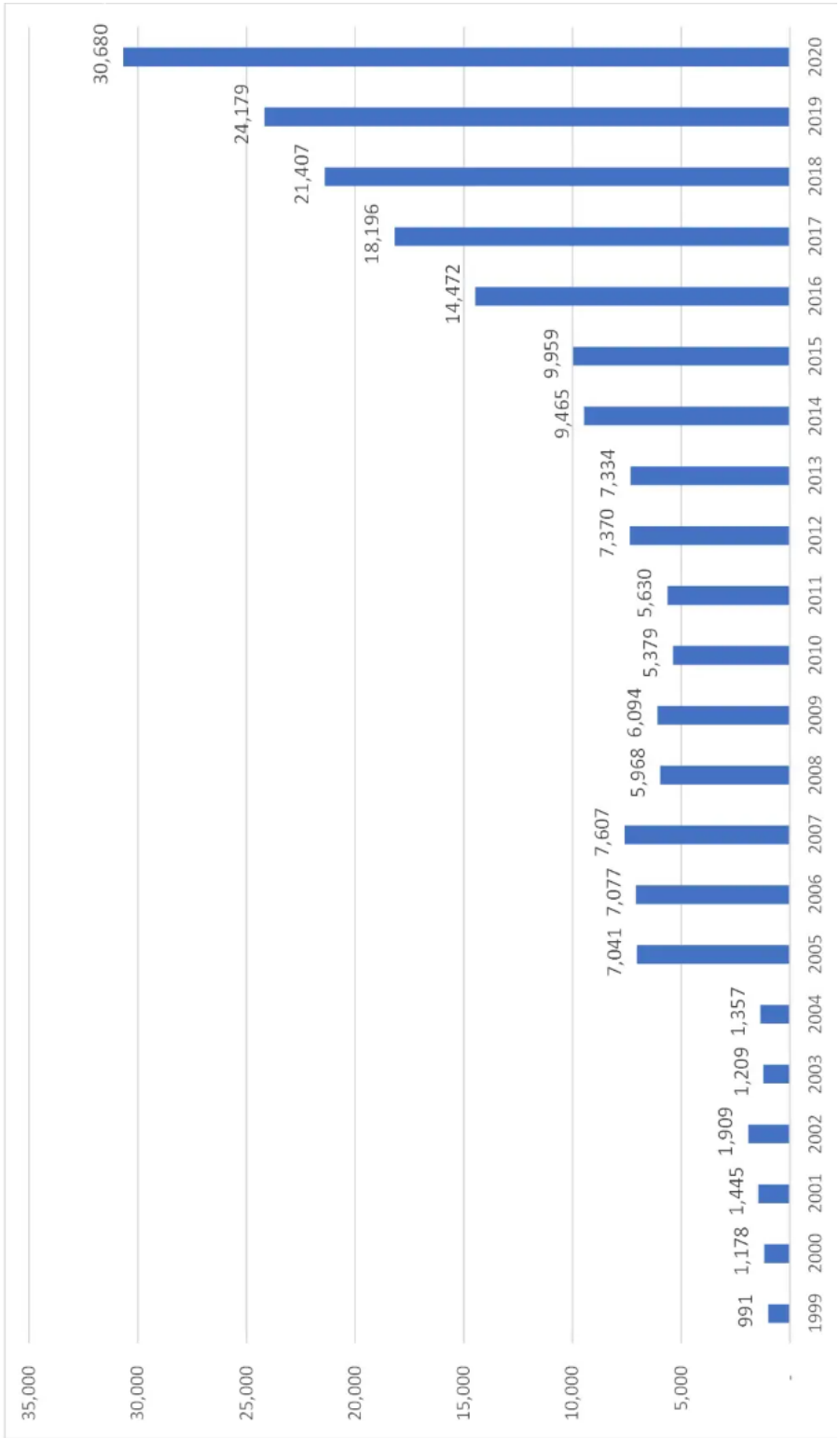


Figure 33: Reported vulnerabilities by year (Reproduced from [5]).

Figure 34: Vulnerability management process lifecycle (Reproduced from [4]).

- **Discover** The first process involves taking an inventory of all assets across the environment, identifying details including operating system, services, applications, and configurations to identify vulnerabilities.
- **Prioritize** Second, discovered assets need to be categorized into groups and assigned a risk-based prioritization based on criticality to the organization.
- **Assess** Third is establishing a risk baseline for your point of reference as vulnerabilities are remediated and risk is eliminated. Assessments provide an ongoing baseline over time.
- **Remediate** Fourth, based on risk prioritization, vulnerabilities should be fixed (whether via patching or reconfiguration). Controls should be in place so that remediation is

completed successfully and progress can be documented.

- **Verify** Fifth, validation of remediation is accomplished through additional vulnerability scanning and reporting.

- **Report** Finally, the organization's management must understand the current state of risk around vulnerabilities. IT needs tactical reporting on vulnerabilities identified and remediated by comparing the most recent scan with previous ones.

The CVSS score is also often used as a metric for risk, despite it not being designed for this purpose. There are actually several public examples of prioritization strategies based on CVSS scores. For example, the US Federal government specified with QTA0-08-HC-B-0003 reference notice that IT products used to manage and assess the security of IT configurations must use the NIST certified Security Content Automation Protocol (SCAP) protocol which explicitly states:

> *"Organizations should use CVSS base scores to assist in prioritizing the remediation of known security-related software flaws based on the relative severity of the flaws" [129].*

A policy from FIRST suggests patching everything with CVSS 7.0 and above [140]. Another notable example is PCI DSS, the standard for security of credit card data that states a similar rule based on a predetermined scoring range:

> *"Risk rankings should be based on industry best practices. For example, criteria for ranking high risk vulnerabilities may include a CVSS base score of 4.0 or above"* [127].

As a result, the CVSS base score is commonly used in the industry to identify 'high risk' vulnerabilities that must be fixed with the highest priority. However, it is not clear whether this interpretation of the CVSS score matches with attacks in the wild. A key issue is whether the critical score actually matches the risk of exploitation in the wild, and if so, how that score can be improved.

2.5 LIMITATIONS OF VULNERABILITY DATA

Quality issues in vulnerability databases, e.g., NVD, have been previously noted and studied. A general overview of these problems is given in Christey and Martin [28]. Prior work has investigated certain types of data quality concerns in the NVD. Anwar et al. [13] performed an in-depth large-scale analysis of the NVD, systematically evaluating each data

field it contains. In particular, they identified significant data issues with the vulnerability publication date, affected vendor and product names, severity scores, and vulnerability type. Anwar et al. [13] also identified that the CWE field for CVE-IDs is not consistently populated with a specific CWE-ID value (Figure 35).

Weakness Enumeration

CWE-ID	CWE Name	Source
NVD-CWE-Other	Other	NIST

Figure 35: The NVD-CWE-Other can be associated with a CVE-ID but it is not included in the CWE repository (Reproduced from [100, 114]).

The NVD is also subject to inconsistent product versions, as demonstrated by Nguyen and Massaci [112]. Nguyen and Massaci [112] pointed out that the affected product versions in the NVD are often incorrect, observing that roughly 25% of Google Chrome CVEs had an incorrect Chrome version string. In the NVD, a CVE should be assigned a vulnerability type under the CWE category to provide users with an overview of the vulnerability nature and risk. Fitzgerald and Foley explained that the CPE naming specification does not offer a mechanism to define relationships between CPE IDs, and therefore, ambiguities are likely to exist [54]. Christey and Martin [28] similarly explored issues in the NVD data and suggested reporting biases as a root cause. Attila et al. [76] showed that CVSS metrics are more suitable for enterprise software products than personal ones. Dong et al. [47] analyzed the inconsistencies in public security vulnerability reports, including the NVD, and found overclaims and underclaims in the affected software product versions. Dong et al. [47] leveraged natural language processing methods to correct inconsistencies in product versions using the NVD reference URLs. To abate these noted inconsistencies and criticisms about data quality, Johnson et al. [88] estimated the accuracy of CVSSv2 base scoring metrics using a Bayesian statistical model. They concluded that with the exception of a few scoring dimensions (i.e., access complexity, authentication), the NVD is trustworthy and reliable data source.

2.6 CHAPTER SUMMARY

In this chapter, we performed a high-level review of the fundamentals of the cyber data exchange and threat intelligence. We then outlined the foundational technologies and processes involved in vulnerability management. Finally, we gave an overview of cybersecurity fundamentals that are relevant and a prerequisite for exploring the research described in this work. Each of these sub-topics, combined together, serves as the basis on which we built the framework for ranking and prioritizing published vulnerabilities. We also discussed some of the issues involved in studying vulnerabilities and data needed to perform an objective evaluation.

Chapter 3

RELATED WORK

In this chapter we discuss previous research that falls within the scope of techniques for developing a cyber threat intelligence ontology, exploit prediction, and vulnerability. Section 3.1 discusses research related to cyber ontologies which we will build upon while exploring RQ1. Section 3.2 discusses relevant work on vulnerability categorization (RQ2). The final section of this chapter will discuss techniques for vulnerability prioritization (RQ2).

3.1 CYBERSECURITY ONTOLOGIES

To address RQ1, "factors that can be used to model attack vectors and security threats based on the skill level of a cyber adversary and their motivation to target a specific industry domain", we will evaluate different methods used to aggregate the cyber and vulnerability intelligence data sources identified in Chapter 2. In this section, we present related work that creates ontologies which are then used to prioritize and rank vulnerabilities based on the product configuration of a given enterprise network.

Ontologies are concerned with modeling knowledge of a domain through the use of well-defined concepts and relationships. Many public sources of cyber threat and vulnerability information exist as the basis to defend cyber systems. As discussed in Chapter 2, cyber threat intelligence gathering is a field within the domain of cybersecurity which consists of collecting, exchanging, and analyzing threat intelligence to detect, prevent, and attribute cyber attacks. The strategic goal to define collection methods in this domain is relatively new, and recent years have seen a growth in the development of taxonomies and enumerations for describing vulnerabilities, malware, tools, attack patterns, and other categories of cyber threat intelligence. A shared ontology limits complexity and organizes knowledge with a controlled vocabulary. The creation of an ontology is the first step towards processing and automation our vulnerability prioritization scheme.

Hemberg et al. [73] introduced BRON, an open-source, relational graphing tool that links public threat data from MITRE's ATT&CK matrix, CAPEC, CWE, and CVE IDs, all described in Chapter 2. The goal of BRON is to provide a conveniently connected graph of multiple public sources in the domain of cybersecurity (Table 3) that are already connected but challenging to traverse coherently.

Table 3: BRON information sources and types, organization, and short descriptions. Derived from Hemberg et al. [73], Table 1.

Type of Entry	**Source**	**Description**
ATT&CK **Tactics**	MITRE	12 common tactics of attack staging as shown in the columns of the ATT&CK matrix.
ATT&CK **Techniques**	MITRE	Means of achieving a tactical objective, organized by Tactic, the row elements of the ATT&CK matrix. In BRON, a Technique is a child of a Tactic.
CAPEC **Attack Patterns**	MITRE	Relational concept linking to a Technique (parent) and/or Weakness (child). Relates to abstract how and why (Tactic and Technique) of an attack objective and/or to abstract "where" (Weakness) target of the attack.
NVD CWE Common **Weakness** Enumeration	NIST	Security-related flaws in architecture, design, or code.
NVD CVE Common **Vulnerabilities and Exposures**	NIST	Security-related flaws in software and applications.
NVD CPE entry field: **Known Affected Product Configuration**	NIST	Specific software application or hardware platform releases that are affected, given a parent Vulnerability. Identified using the CPE naming specification.
NVD CVE entry field: Severity score	NIST	Severity is scored using the CVSS. A higher score denotes greater the impact of a Vulnerability.

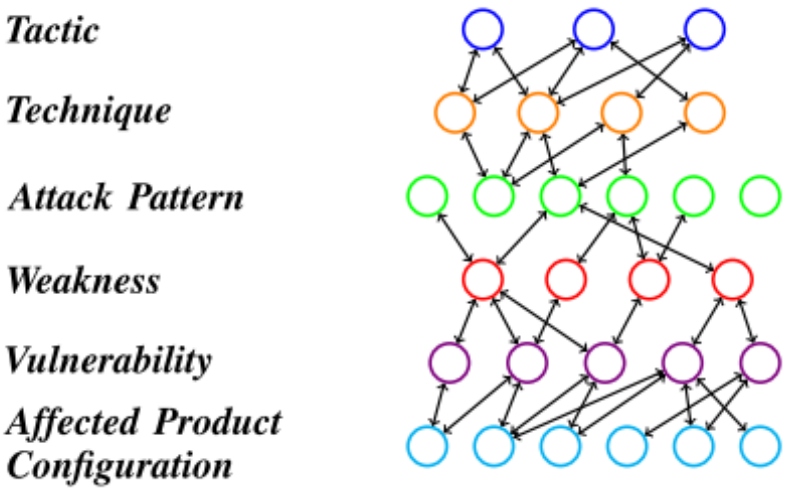


Figure 36: Schematic of BRON's graph of the combined sources (Reproduced from [73]).

The schematic for BRON shown in Figure 36 preserves all entries and relations while enabling bi-directional, relational path tracing. Source and list entries are nodes and relational links are edges. BRON exploits attack patterns to trace between the objectives and means of attacks to the vulnerability. Hemberg et al. fuse the enumerated data sets to create a relational graph and facilitate ease of use. They also collected products of a vendor by parsing the CPE notation for Affected Product Configurations to extract the vendor, product, and version. Specifically, ATT&CK provides the tactics and techniques that attackers could employ against a vulnerable system. The CWE, CVE, and CPE provide information concerning where those weaknesses and vulnerabilities exist in a system based on the affected product configuration. Finally, attack patterns link potential attack actions to weaknesses that could become attack targets. Hemberg et al. also observed what they call *Floating Entries* that are disconnected from the rest of the graph. The *Floating Entries* include attack classifications from CAPEC with no indication of how they could be accomplished or what weakness they would exploit. No Weaknesses or Affected Product Configurations are Floating Entries. However, of 71,715 vulnerabilities in the Hemberg et al. data set, 28% are Floating Entries. Also, of the 519 Attack Patterns, 25% are Floating Entries. This draws attention to a level of in-completion not only within BRON, but the challenge in achieving currency even with standardized data sets that are vetted and well maintained.

Table 4: Groups are sets of real-world intrusion activities that are tracked by a common name in the security community. Industry targets are noted in bold. Extracted from the MITRE ATT&CK matrices [107].

ID	Name	Description
G0005	APT12	APT12 is a threat group that has been attributed to China. The group has targeted a variety of victims including but not limited to **media outlets, high-tech companies, and multiple governments.**
G0037	FIN6	FIN6 is a cyber crime group that has stolen payment card data and sold it for profit on underground marketplaces. This group has aggressively targeted and compromised point of sale (PoS) systems in the **hospitality and retail sectors.**

Our research differs from BRON in that we extracted additional information from the MITRE ATT&CK matrix. Specifically, we collected the threat groups and the known techniques used to initiate a cyber attack. The MITRE ATT&CK matrix uses the term *Group* as their designation for a cluster of adversary activity. A subset of the group designations is shown in Table 4. Text mining of the group descriptions revealed the preferred target industries for each group. Particular emphasis is placed on identification of preferred industry targets needed to address RQ2.

Iannacone et al. [78] aimed to create an ontology that can represent the cyber security domain, allowing information to be combined from as many sources as possible within this domain. Their Situation and Threat Understanding by Correlating Contextual Observations (Stucco) [72] ontology and knowledge graph incorporates information encompassing 115 properties and provides relationships among 15 entity types (Figure 37) including software, vulnerabilities, and attacks. Security event data from Intrusion Detection System (IDS) alerts provide a starting point for their analysis. To provide context for a particular security event, Iannacone et al. manually gathered and synthesized relevant data from system logs, network flows, and firewall data. As shown in Figure 37, the referenced entities and relationships were gleaned from IP blacklists and reputation lists, software vulnerability information, malware and threat data, operating system and application vendor blogs, and news sites. Unlike other cyber ontologies, there is no explicit Exploit entity in Stucco; it is

instead grouped with the Malware entity. The use cases for the Stucco ontology are geared towards System Administrators performing incident reporting.

The system administration use cases [78] include:

- Searching through network flow records and IDS records by IP address during some time window, and comparing remote addresses against blacklists or reputation systems
- Gathering information about the software packages on impacted hosts, and comparing with vulnerability databases like the NVD and IDS alerts
- Attempting to identify malware based on system changes and network traffic logs

The authors aimed to integrate information from both structured (e.g., JSON) and unstructured data sources (e.g., text articles, blogs). The Stucco ontology provides a framework for identifying entity types and properties which represent needed fields contained in cyber data sets. We leveraged their approach for entity extraction [21, 89] as we developed data mappings during the construction of knowledge graphs using the text description fields in the NVD and ExploitDB as a corpus.

In addition to what was already noted in Section 2.5, Kiesling et al. [92] described additional limitations in the MITRE and NIST data sets we discussed in Chapter 2. First, they contend individual entities and data sets remain isolated and cannot easily be referenced and linked from other data sets. Second, while the governed schemas provide a well-defined structure, the semantics are not as well-defined. In response, Kiesling et al. provided tools and services that illustrate how Linked Data [19] principles can be applied to combine local and public knowledge using an Extraction, Transformation, Loading (ETL) pipeline. The complete vocabulary [50] they define for their SEPSES Cybersecurity Knowledge Graph links five well-established standards in the cybersecurity domain, namely the CVE, CAPEC, CWE, CPE, and CVSS. Most of these data sets define unique identifiers for key entities such as vulnerabilities, weaknesses, and attack patterns which are leveraged by SEPSES to link the data as part of the ETL process. For example, a CVE instance may have a relation to another resource such as a CPE identifier. Based on these identifiers, Kiesling et al. create direct links between associated resources. The SEPSES schema shown in Figure 38 integrates the NVD which enriches CVEs with additional information, such as security checklist references, security-related software flaws, misconfigurations, product names, and impact metrics. This is represented in the CVE class which includes data type properties such as cve:cveId, cve:description, cve:issued, and cve:modified timestamps.

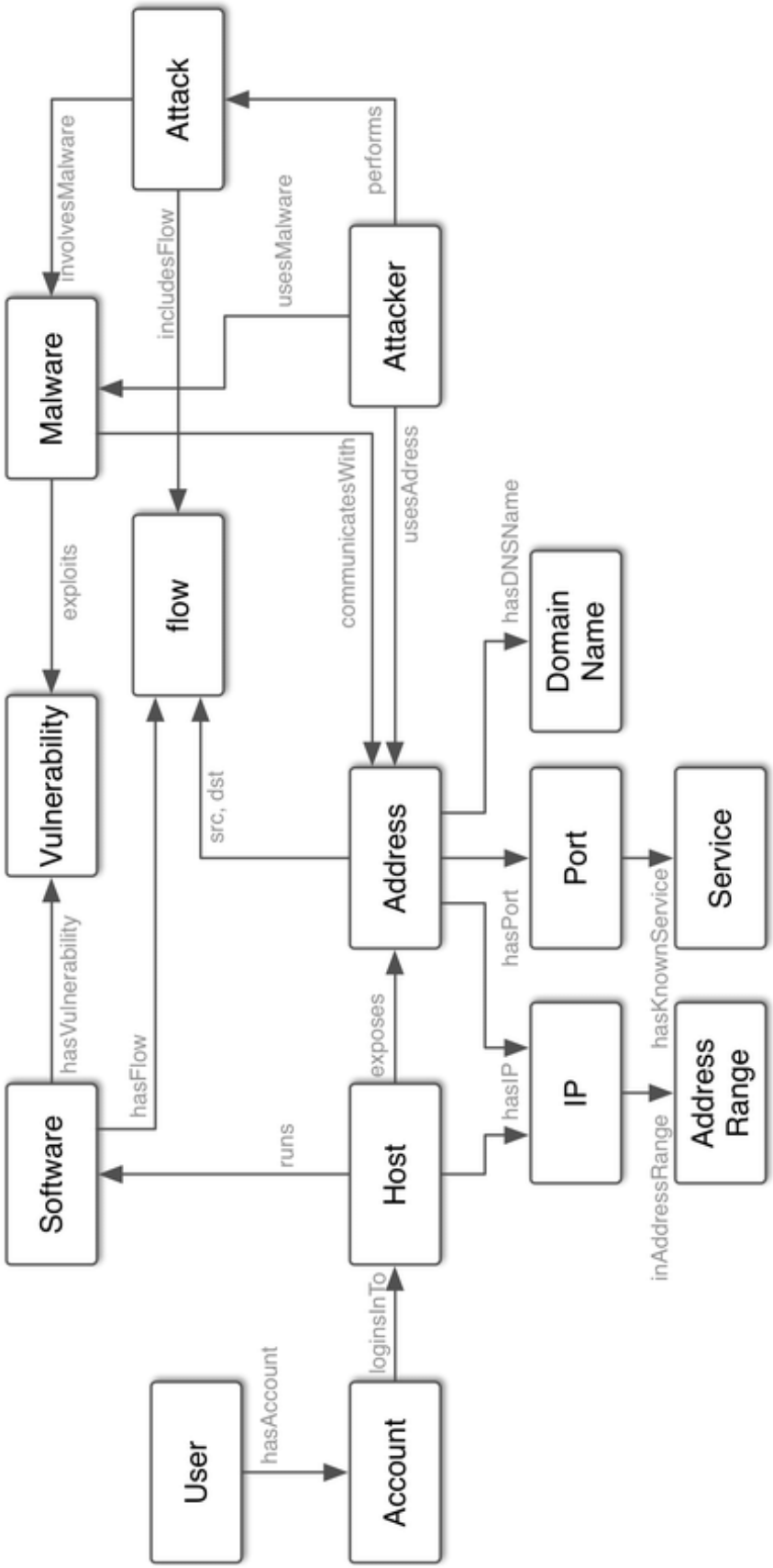


Figure 37: Entities and relations in the Stucco ontology (Reproduced from [78]).

Based on the NVD information, SEPSES can link CVE to affected products (cve:hasCPE), vulnerable configurations, (cve:hasVulnerableConfiguration), impact scores (cve:hasCVSS), related weaknesses (cve:hasCWE), and external references (cve:hasReference). SEPSES uses some of BRON's data sources, however BRON is unique in incorporating the tactics and techniques from the ATT&CK matrix and bridging them to CWE via CAPEC.

Wang et al. [163] developed a vulnerability ontology, called OVM (Ontology for Vulnerability Management), which captures the relationships between IT products, vulnerabilities, attackers, security metrics, countermeasures, and other relevant concepts. Similar to other ontologies, OVM incorporates common standards such as CVE, CVSS, CWE, CPE, and CAPEC. The top level concepts of the OVM ontology shown in Figure 39 include the Vulnerability, IT_Product, Attacker, Attack, Consequence, and Countermeasure. More specifically, a Vulnerability existing in an IT_Product can be exploited by an Attacker by conducting an Attack with the objective to compromise the IT_Product and cause Consequence. A Countermeasure is an action or approach that will protect the IT_Product by mitigating the Vulnerability. The object properties are defined as relations between instances of classes. For example, the relations between IT_Product and Vulnerability are described by the hasAffectedProduct and hasVulnerability object properties. With the OVM, Wang et al. sought to describe the pattern of external threats and internal vulnerabilities formally and precisely. Similar to our research, the OVM provides the foundation for building automated tools which reduce the scope, complexity, and volume of security data that must be managed by security professionals.

3.2 VULNERABILITY CATEGORIZATION

In our work, after creating the requisite cyber ontology, we will need a methodology to categorize the vulnerabilities based upon a common set of characteristics. Categorization is a pre-cursor to developing a remediation strategy. Categorization is also necessary to address RQ2, "What are the characteristics needed to define vulnerability ranking policies that improve the return on investment (ROI) of applied mitigations, compared to traditional CVSS Base Score policies, relevant to the organization's specific mitigation goals and priorities?" We may also be able to identify features present in other cyber threat intelligence libraries from authoritative sources which can be used to improve the vulnerability ranking strategy we propose. Research has demonstrated several different approaches for categorizing vulnerabilities and enriching CVSS scores.

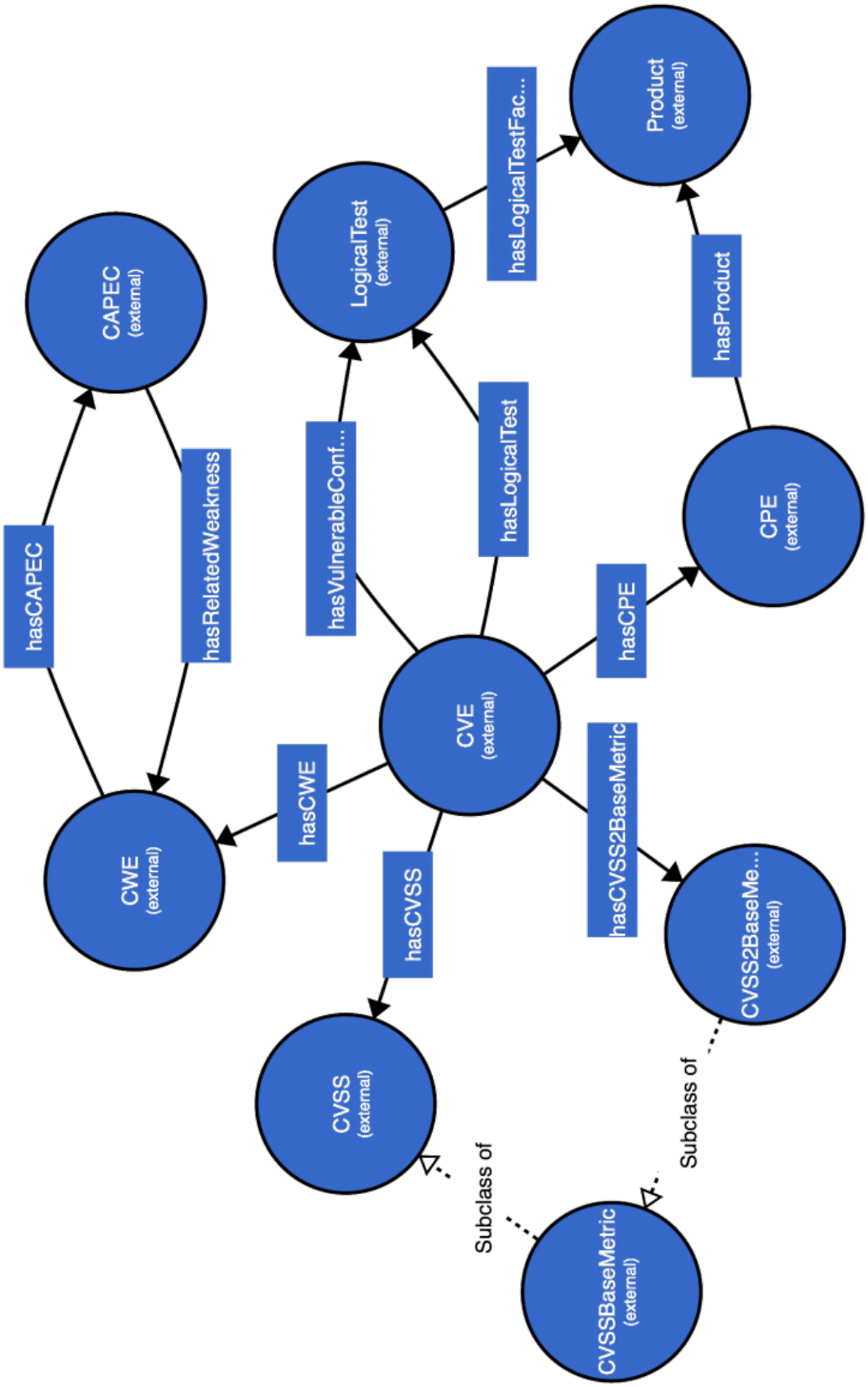


Figure 38: SEPSES knowledge graph vocabulary high-level overview (Reproduced from [92]).

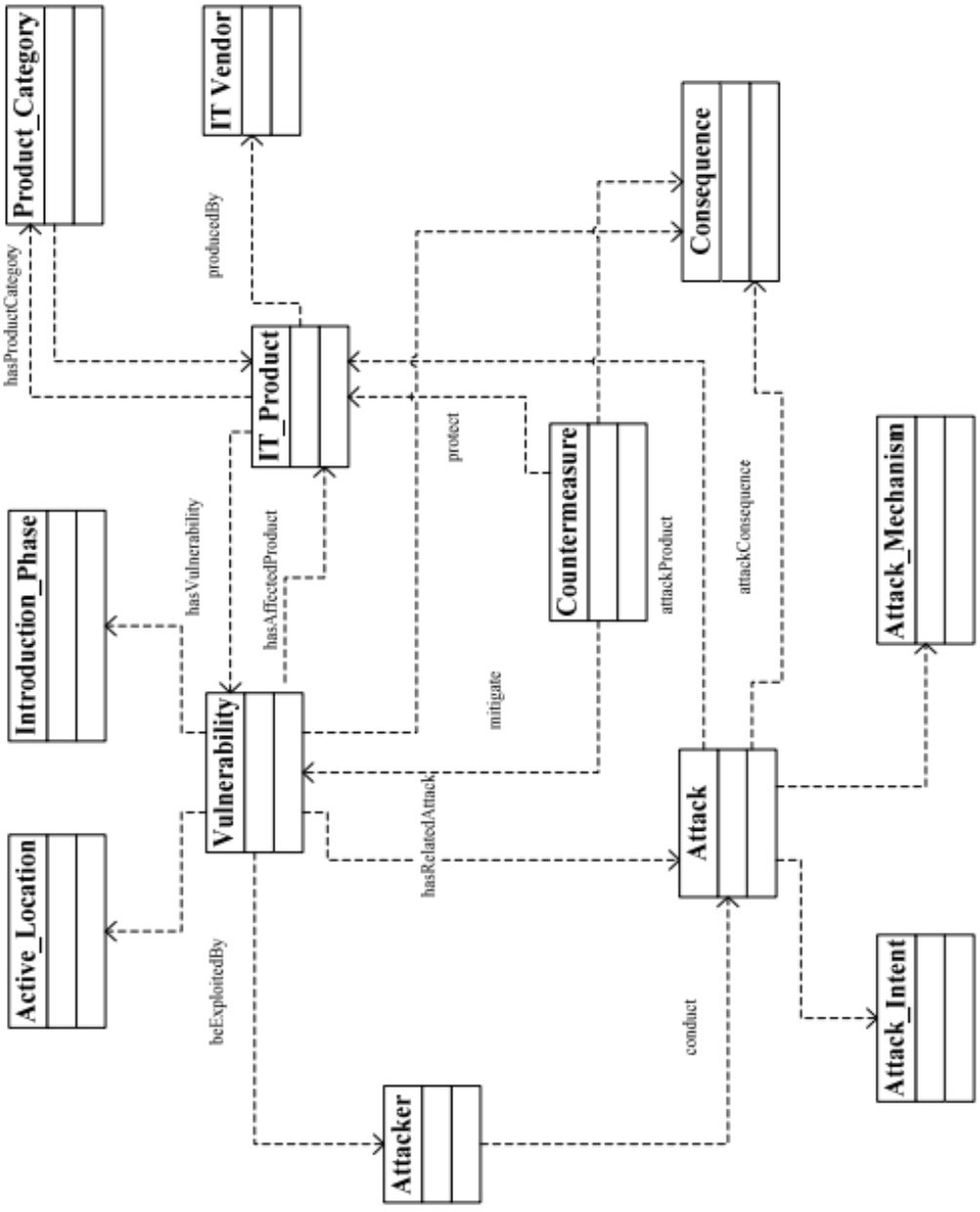


Figure 39: Conceptual model of the vulnerability ontology (Reproduced from [163]).

3.2.1 CVSS SCORING METRICS

The NVD publishes the CVSS base score as a severity indicator for all recorded vulnerabilities. The two context aware CVSS scoring metrics (i.e., Environmental, Temporal), however, are often omitted. Since the base score is unaware of an organization's context, the NVD scoring alone is of limited use for vulnerability prioritization in practice [49, 91]. Fruhwirth et al. [61] developed a set of context-enriched scores that apply the CVSS temporal and environmental scoring equations. They simulated the missing context information (e.g., the availability of vendor patches or exploits) using a trend analysis employed by Frei's Pareto distribution [59] based on the CVSS disclosure date and the Environmental (unique contextual) metrics based on the empirical, interview-based data from the security managers. Using scenarios to represent an organization's investment in security operations, they applied the presented method on a set of 720 actual vulnerability samples published in the NVD and found that the application of context information had a significant impact on vulnerability scoring. In general, it led to a reduction of the average CVSS score values by about 0.5 points. Even though this reduction was small on average, it had a severe impact on the subsequent classification of vulnerabilities by downgrading many vulnerabilities that were formerly considered critical. In Figure 40, 131 vulnerabilities were originally classified as critical using the CVSS Base Score alone. Conversely, only 31 were similarly classified using environmental scoring. This work supports our inclination that vulnerability scores can be improved with additional information based on the organization's perspective. In lieu of the simulation approach used by Fhurwith et al., we will take advantage of vendor advisories published in the NVD to calculate enhanced temporal and environmental scores. To measure an organization's investment in security, Fhurwith et al. also implemented a factor, vice absolute monetary value, that represents the total costs required to resolve a vulnerability in a particular class. They assigned the following example cost factors to the vulnerability classes: Low: 0.25, Medium: 1, High: 1.50, Critical: 3.00. To determine the total cost of resolving all vulnerabilities, the cost-factors are multiplied by the number of vulnerabilities in their corresponding class. This method of measuring a return on security investment can contribute to evaluation of the ranking strategy we propose.

Severity Class (cost factor)	**Scenario A** CVSS Basic Score only # of Vuln	costs	**Scenario B** CVSS Score with Context # of Vuln	costs	**Difference** #	costs
Low (0.25)	38	10	121	30	+83 (+218%)	+21
Medium (1)	248	248	171	171	-77 (-31%)	-77
High (1.5)	303	455	397	586	+94 (+31%)	+141
Critical (3)	131	393	31	93	-100 (-76%)	-300
Total	***720***	***1105***	***720***	***899***		***-215 -19%***

Figure 40: Severity classifications of vulnerabilities in the analyzed sample set with and without the application of context information that applies the CVSS's temporal and environmental metrics (Reproduced from [61]).

Another aspect of CVEs often considered during remediation decisions is the CVSS score, which is included for over 90% of all CVEs in the NVD. Based on a scale of 0 to 10, CVSS scores reflect assessments of the underlying vulnerability characteristics. We consider the table of the overall scores in Figure 41. A review of the vulnerability counts partitioned between the red lines shows it is apparent that 77% of CVEs fall in the 4 to 8 range for CVSS scores. This observation is somewhat counter intuitive with patching strategies mentioned previously that prioritize vulnerabilities in the 9 to 10 range (i.e., high, critical).

CVSS Score	Number Of Vulnerabilities	Percentage
0-1	848	0.50
1-2	1092	0.70
2-3	6947	4.50
3-4	7080	4.60
4-5	36120	23.20
5-6	29887	19.20
6-7	22319	14.30
7-8	32072	20.60
8-9	740	0.50
9-10	18455	11.90
Total	155560	

Weighted Average CVSS Score: **6.5**

Figure 41: CVSS score distribution for all vulnerabilities (Reproduced from [126]).

3.2.2 TEXT MINING

Jacobs et al. [84] describe a methodology for creating a decision model which most efficiently prioritizes the exploit potential of vulnerabilities. They classify vulnerabilities as either high or low risk, where a high risk vulnerability is considered one that has been exploited in actual industrial networks. They aggregate data from a variety of sources, including a private data set generated by Kenna Security,[1] a firm that monitors more than 100,000 corporate networks (almost 200 billion observations of real-world intrusion detection systems). The model uses text mining to extract 191 tags (e.g., buffer overflow, denial of service) from the detailed description of vulnerabilities published between 2009 and 2018. The tags are included as features in their prediction model which uses gradient-boosted

[1] `https://www.kennasecurity.com/`

decision trees [26] to build a predictive model.

Almukaynizi et al. [11] mined the NVD description to glean information on the vulnerability and the capabilities attackers will gain if they exploit it. Contextual information gleaned from the dark web was appended to the NVD description. Here, the authors observed foreign languages in use which they translated into English using the Google Translate API. The text features were analyzed using Term Frequency-Inverse Document Frequency (TF-IDF) to create a vocabulary of all words in the entire data set which they subsequently limited to 1000 most frequent words. Common words were eliminated as important features. Almukaynizi et al. employed several supervised machine learning approaches to determine a binary classification on the selected features indicating whether the vulnerability would be exploited or not. The model shown in Figure 42 extracts feature information from the NVD, ExploitDB, a vulnerability detection community (Zero Day Initiative), and forums on the dark web.

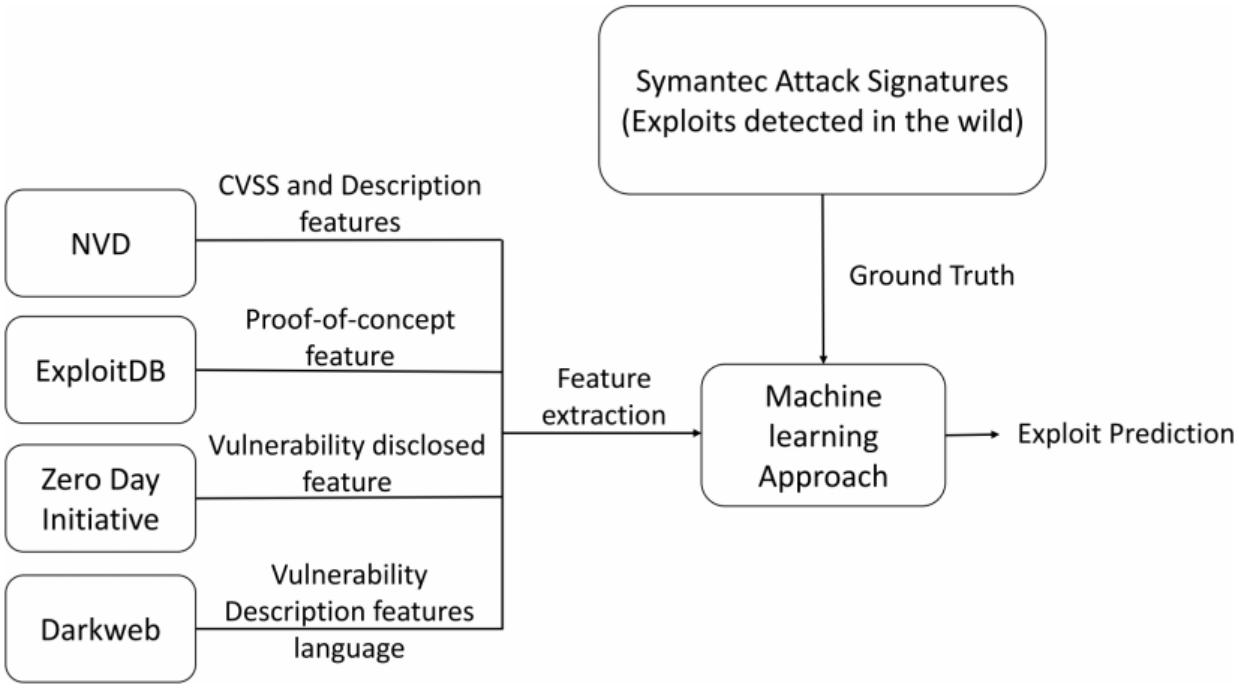


Figure 42: Exploit prediction model (Reproduced from [11]).

Conversely, Khazaei et al. [90] used a similar text mining approach to extract feature vectors from the now deprecated Open Sourced Vulnerability Database (OSVDB) [165]. They were able to predict the CVSS score with 88% accuracy based on the vulnerability description.

3.2.3 BUSINESS/MISSION CONTEXT

Internal stakeholders can provide input concerning the enterprise architecture itself and how the system affects the organization's culture, operations, and strategy. Alberts and Dorofee [6] observed that many risk evaluation methods do not review and analyze risks to an organization's mission and business strategies. The Operationally Critical Threat, Asset, and Vulnerability Evaluation (OCTAVE) [157] threat model they developed helps an organization understand its most important information, technology, and other assets, and what risks can threaten those assets. OCTAVE is a self-directed approach, meaning that people from the organization assume responsibility for setting the organization's security strategy. As a result, most of the risk evaluation in OCTAVE is qualitative, based on domain expertise and knowledge of the participants.

3.3 VULNERABILITY PRIORITIZATION

This section addresses RQ2 as we seek to define an approach to prioritize security controls or software patches that can be applied to close the vulnerabilities and prevent potential cyber attacks. We aim to identify which security threat (i.e., vulnerability) is more critical than other identified threats based on the skill and expertise of the adversary. Most work in the area of exploit prediction focuses on vulnerabilities with a published exploit as opposed to an exploit observed in the wild. This is because predicting actual exploits in the wild remains a difficult prediction problem. Sabottke et al. found that less than 1.4% of disclosed vulnerabilities are observed in the wild [134]. Published exploits (e.g., ExploitDB), however, are the most relevant proxy for the probability that an exposed vulnerability can be used to compromise a network. As a result, a significant amount of related work is focused on building predictive models that can identify an exploited vulnerability.

3.3.1 CVSS SCORING METRICS

A critical challenge for many organizations is understanding how to minimize the cost of managing and protecting its information assets and business systems. A core component of

this challenge is adopting a vulnerability management process that can detect and remediate known vulnerabilities [9]. A common approach is to remediate all vulnerabilities above a certain severity score. However, many of the common approaches used by organizations have been found to be sub-optimal [46] and in some cases, no better than randomly choosing vulnerabilities to remediate [9]. It may be infeasible to patch even CVEs with the highest CVSS base scores due to the time and resources required for remediation actions. Alperin et al. noted that 13.5% of the NVD vulnerabilities are scored between 9 and 10 [12].

Allodi and Massacci [8] examined whether the CVSS score can be a good predictor for vulnerability exploitation, and whether it can be improved by additional information. They proposed the case-control study methodology as an operative framework for security studies. In a case-control study, the researcher looks backward at some of the cases (e.g., vulnerabilities exploited in the wild) and compares them with controls (in their case, randomly selected vulnerabilities with similar characteristics, such as year of discovery or software type). The goal is to identify whether some risk factor (e.g., a high CVSS score, the existence of an exploit kit) provides a good explanation of the cases and therefore represents a decision variable upon which the System Administrator can act.

To illustrate the methodology, they first analyzed how the CVSS score expresses the impact and the likelihood of an exploitation to happen. Allodi and Massacci showed that a proper characterization of "likelihood of exploit" is not present in the CVSS score. Next, they evaluated its performances as a risk indicator by performing a case-control study that tests how the CVSS score correlates with exploitation in the wild. Their results showed that the CVSS base score never achieves high rates of true positives (sensitivity) simultaneously with a high rate of true negatives (specificity). An analysis of the distribution of CVSS scores and subscores was presented by Mell et al. [103] and Gallon [62]. However, while including CVSS subscore analysis, their results are limited to data from NVD and do not specifically address vulnerability exploitation.

Feature extraction from public resources has also been actively explored. Blogs, user forums, and other public web resources could contain exhaustive information about security issues. In addition to mining CVE descriptions, Tatarinova et al. [152] used the Wayback Machine [2,119] and Google Trends [1] to perform a temporal analysis of changes to vendor references found in the NVD. A comparison of user-generated content related to security vulnerabilities on digital platforms (Reddit, Twitter, GitHub) was studied by Horawalavithana et al. [75]. Their analysis showed that while more security vulnerabilities are discussed on social media sites like Twitter, relevant conversations go viral much earlier on Reddit.

They also concluded both Reddit and Twitter could be used to accurately predict peaks in software development activity on GitHub when a CVE-ID is explicitly mentioned.

3.3.2 LIKELIHOOD OF EXPLOIT

Ross et al. [133] sought to improve upon the CVSS score ranking by exploring the latent feature space described by a Jaccard similarity metric. Their goal was to provide a data-driven and alternative ranking approach using features in the CVSSv2 base and temporal metric groups in the NVD. The authors used four data sets reporting data on vulnerabilities and CVSS scores, proof-of-concept exploits, exploits traded in the cyber black markets, and exploits in the wild. Ross et al. provided a data-driven analysis to present improvements to existing vulnerability ranking systems by tying exploits to the risk presented by the vulnerability.

The authors were also motivated to analyze the NVD's latent feature space by observing the emergence of CVSSv2 vectors. In particular, they determined 80% of the NVD entries in their dataset could be described using only 17 vectors. Using spectral and k-means analysis, Ross et al. observed that three clusters suitably captured the CVSS vector feature spaces within the NVD. Finally, the authors concluded their rank-ordering of CVSS vectors is superior to that implied by CVSS scores when exploits are used as a stand-in for risk. To provide validation of their algorithm's accuracy over time, Ross et al. observed vulnerabilities and exploits for a period of time, built a model, and then measured the emergence of new exploits at intervals of six months, one year, and two years.

The Ross et al. clustering strategy, where k is the number of clusters, provides a mechanism to prioritize patching by examining the likelihood of an exploit existing for CVE-IDs with similar CVSS vectors. They also observed similar performance with traditional ranking methods once six clusters are mitigated. This observation indicated that low-scoring vulnerabilities and low-ranked clusters have few exploits.

Jacobs et al. [84, 86] developed a data-driven framework for assessing whether a vulnerability will be exploited in the wild within the first twelve months after public disclosure. Their binary outcome model, Exploit Prediction Scoring System (EPSS), used logistic regression to inform feature selection from the NVD, CPE, and CVSS base score. Of the final variables selected, seven related to the software vendor, two related to exploit code, and six related to descriptive properties of the vulnerability and impact in a published CVE ID. The final set of 16 variables and the associated scoring equation was then compared to a patching strategy based on the CVSS base score (column 1 in Figure 43). These results

suggest that a rule-based strategy of remediating all vulnerabilities with CVSS 7 or higher would achieve coverage of slightly over 74% with an efficiency of 9%, and accuracy of only 57%. This appears to be the best balance among CVSS-based strategies, even though it could still result in unnecessarily patching unexploited vulnerabilities. While a stated of goal of Jacobs et al. was to create an open data model, they utilized a proprietary data set, provided by a security vendor, that contains vulnerability exploits (i.e., intrusion detection signatures).

Almukaynizi et al. [11] draw on a body of work that seeks to define an exploit prediction model that leverages data from online sources generated by the white-hat community (i.e., ethical hackers). The white-hat data is combined with vulnerability mentions scraped from the dark web to provide an early predictor of exploits that could appear in the wild (i.e., real world attacks). Almukaynizi et al. assessed the importance of aggregating disparate data sources by first analyzing the likelihood of exploitation based on the coverage of each source (Figure 42). Then, they conducted a language based analysis to identify any socio-cultural factors present in the dark web sites which might influence exploit likelihood. Experiments using the exploit prediction model were examined using different supervised machine learning algorithms including Support Vector Machine, Random Forest, Naive Bayes Classifier, and Logistic Regression. Random Forest, a technique which generates multiple decision trees, was found to provide the best F1 score to determine classes of exploited versus not exploited vulnerabilities. Their classifier was evaluated based on precision, recall, and Receiver Operating Characteristics [174].

Almukaynizi et al. also included a language analysis of the vulnerability mentions extracted from their online data sources and exploitation likelihood. Figure 44 shows English and Chinese have more vulnerability mentions (n = 242, and n = 112, respectively) than Russian and Swedish (n = 13, and n = 11, respectively). However, vulnerabilities mentioned in Chinese postings exhibited the lowest exploitation rate. Although vulnerability mentions in Russian or Swedish postings are lower, these vulnerabilities exhibited very high exploitation rates. For example, about 46% of the vulnerabilities mentioned in Russian were exploited (n = 6), and about 19% for vulnerabilities mentioned in Swedish (n = 2). The correlation of exploit likelihood to languages is similar, but different from our approach to link vulnerabilities to an adversary using their country association found in the MITRE ATT&CK matrix.

Strategy	Accuracy (%)	Efficiency (%)	Coverage (%)	Level of effort (# vulns)	Efficiency by chance (%)	Coverage by chance (%)
CVSS 10+	90.40	16.80	18.80	4686	5.53	6.20
CVSS 9+	84.70	17.80	48.70	11 477	5.53	15.20
CVSS 8+	72.40	12.10	63.60	21 990	5.53	29.10
CVSS 7+	57.00	9.00	74.30	34 530	5.53	45.70
CVSS 6+	51.80	8.30	76.60	38 674	5.53	51.20
CVSS 5+	34.10	6.90	87.10	52 906	5.53	70.00
CVSS 4+	10.20	5.70	98.50	71 908	5.53	95.10

Figure 43: CVSS prediction results using a rule-based remediation strategy based on the CVSS base score. Note: Efficiency by chance and coverage by chance are the overall results from a strategy of randomly selecting vulnerabilities to remediate (Reproduced from [84], Table 2).

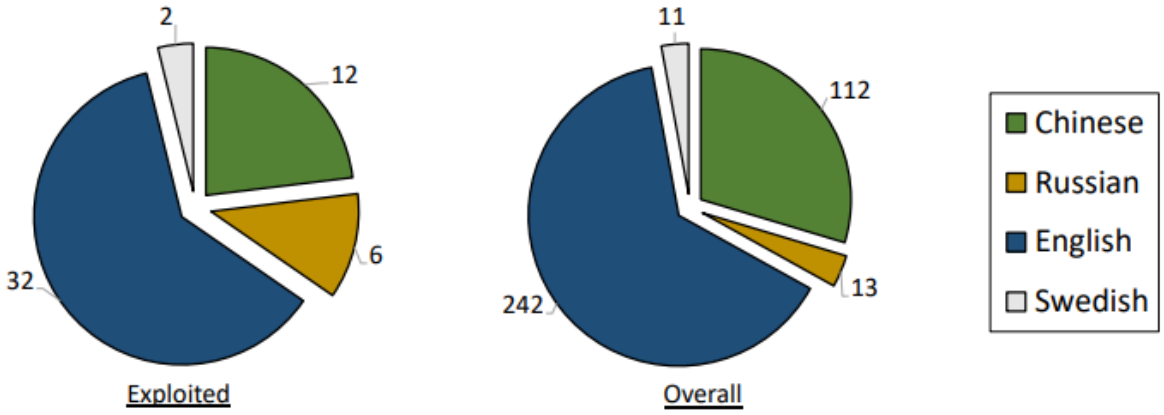


Figure 44: Number of the exploited vulnerabilities mentioned by each language (left), and number of vulnerabilities mentions in each language (right) (Reproduced from [11]).

Some vulnerabilities are never exploited in the wild, partly due to security technologies that make exploiting them difficult. New security metrics were proposed by Nayak et al. [110] which examine the exploitation ratio of vulnerabilities in their deployed environment. Their study focused primarily on vulnerabilities in variants of the Microsoft Windows operating system and on several applications that commonly run on that platform (i.e., Microsoft Office, Internet Explorer, Adobe Reader). Nayak et al. used the NVD, OSVDB, Symantec signatures, and the Worldwide Intelligence Network Environment (WINE) [48] to define field-measurable security metrics to:

- identify which vulnerabilities are exploited (i.e., count of exploited vulnerabilities and exploitation ratio)
- quantify how often they are exploited (i.e., attack volume and exercised attack surface)
- study when vulnerabilities are exploited

CVSSv3 now includes a temporal vector and score that uses an "exploit code maturity" field spanning severity of "unproven", "proof-of-concept", "functional", and "high". However, these fields are not always employed, as the temporal vector must be calculated manually for every vulnerability.

3.3.3 ATTACK GRAPHS

Metrics indicating inherent risk in a network system can help in prioritizing resources to improve security and reduce the possibility of mission interruption from successful cyber attacks. Quantifying a security level for large-scale networks can be challenging. System Administrators currently respond based on their experience rather than objective metrics and models. An attack graph is a model for analyzing security of a network by modeling the way attackers combine and exploit vulnerabilities in a network to achieve their attack goals. Attack graphs can show the cumulative effect of vulnerabilities throughout a network by visualizing the logical dependencies between the adversary's initial position and the adversary's ultimate goal. The paths that an attacker can follow to reach a specific system or resource can be modeled where each node is a vulnerability on a system, and edges represent the attacker's possibility to escalate to the next system [164]. The graphs represent system states using a collection of security-related conditions, such as the existence of vulnerability on a particular host or the connectivity between different hosts. In Figure 45, the left side depicts the configuration of a network. The right-hand side shows the attack graph, which is a directed graph with two kinds of vertices, namely, exploits shown as predicates inside ovals and conditions shown in plain text. A directed edge from a condition to an exploit means executing the exploit requires the condition to be satisfied. The edge from an exploit to a condition means executing the exploit will satisfy the condition. The numerical value inside each oval is a probability that indicates the relative likelihood of the corresponding exploit being executed by attackers when all the required conditions are already satisfied.

Vulnerability exploitation is modeled as a transition between system states [147]. These graphs often tend to be unwieldy as network size grows, making the identification of realistic paths to compromise difficult to achieve [121]. Prior work has been conducted in the area of attack graph construction from network configurations in order to analyze network security [96,144]. Ou et al. [124,125] developed an attack graph generation tool, Multihost Multistage Vulnerability Analysis (MulVal), which is then used to aggregate vulnerability metrics from the NVD [74]. Their approach expressed security semantics as logical rules and system configuration information as a tuple they created for analysis using the CVE ID, and other elements which include the CVSS scoring metrics, attack range of the vulnerability (either remote service, local or remote client), and consequences (i.e., compromises confidentiality, integrity, or availability). Some attack graph methodologies aim to discover individual attacker goals but fail to identify scenarios where vulnerabilities are used for multi-stage attacks.

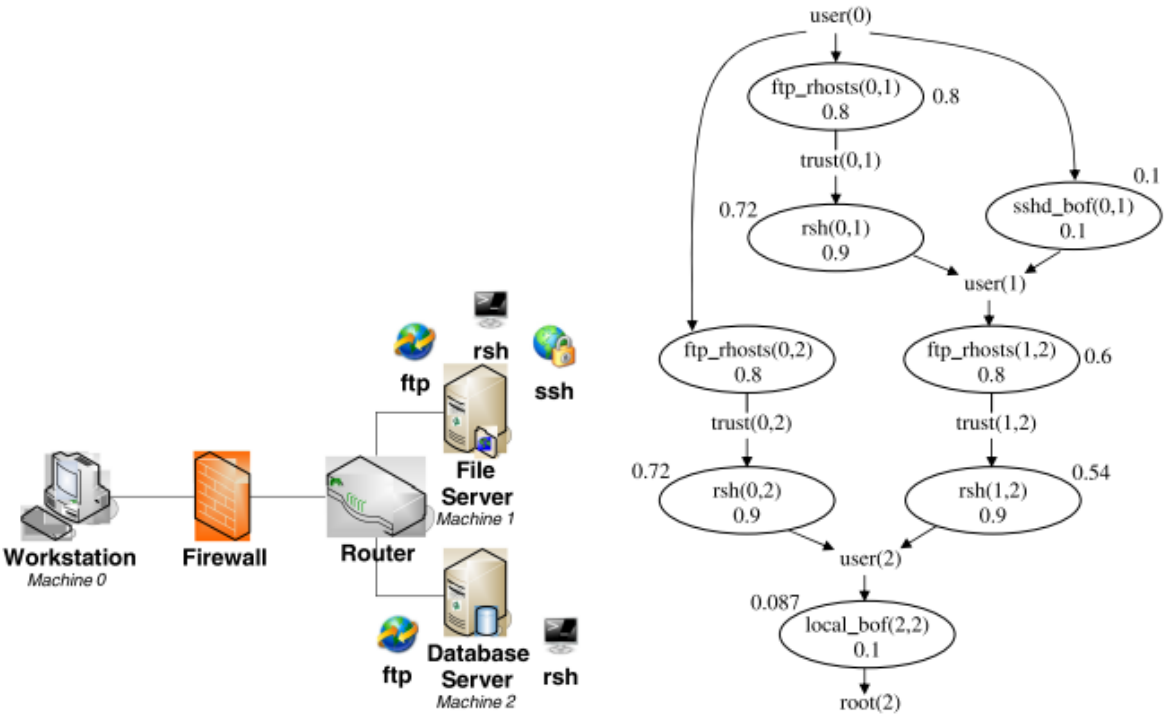


Figure 45: An example of network configuration and attack graph (Reproduced from [164]).

High computational complexity also makes it hard to apply these methodologies in a large scale network. Gallon et al. [64,65] used the CVSS framework with an attack graph to compute the severity level, no longer of an atomic attack targeting one vulnerability, but of a multi-staged attack scenario focused on the CVSS exploitability metrics. The privileges needed to exploit a vulnerability are assessed through exploitability metrics, namely AV (access vector), AC (access complexity) and AU (authentication), and the Exploitability Score (ES). ES is computed based on the assumption the user environment is safe, i.e., the attacker has not previously gained any privilege on the target host. Gallon et al. take into account, in the exploitability vector of each atomic attack, the privileges gained previously by the attacker as an enabler for the next attack path. To reduce or eliminate known limitations previously discussed, we will investigate a property graph model [131]. As shown in Figure 46, a property graph contains nodes, vertices, and relationships that are not only connections but also carry a name (type) and some properties. A property graph excels at showing connections among data scattered across large, diverse data schemas, like those discussed in

Chapter 2, as well as how different kinds of metadata relate. Noel et al. [117, 118] presented a property graph modeling and analytical framework for tracing cyber-attack vulnerability paths through networks, correlated with observed security events. While they incorporate similar MITRE and NIST data sets, the focus is on network penetration and measuring attacker movements using alerts and sensor logs.

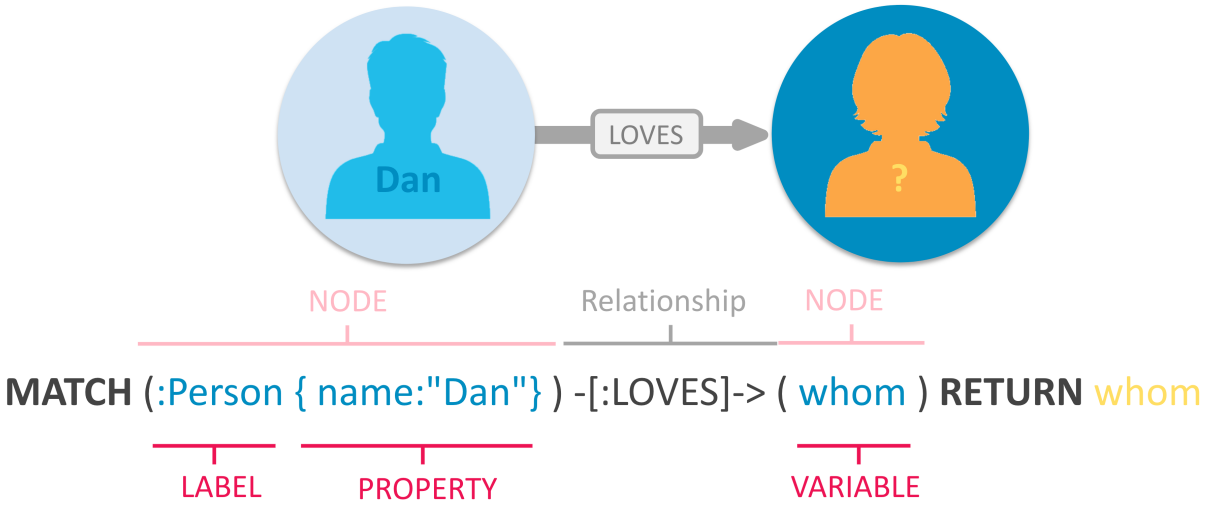


Figure 46: Building blocks of the property graph model (Reproduced from [111]).

3.3.4 ADVERSARY INTENT

To address RQ2, "can we prioritize vulnerabilities based on the skill and expertise of the adversary?" we examine prior work which creates an attacker model in conjunction with machine learning to develop a vulnerability management approach. Alperin et al. [12] proposed a mechanism that customizes vulnerability risks according to their exploitation likelihood in a contested environment given site-specific threat intelligence information, namely, attacks by an Advanced Persistent Threat (APT) group. They assert that attacker models for different adversaries can be used to customize risk prioritization in near-real time, allowing defenders to better visualize their network from an attacker's point of view. This fluidity allows operators to gain situational awareness of their network's vulnerabilities based on

specific threats. Alperin et al. perform NLP on vulnerability descriptions and two security blogs related to APT28, a Russian cyber espionage group also called Fancy Bear [170], and the vulnerabilities they have exploited in the past.

In most attack graph approaches, the vulnerability plays a central role in the threat modeling. The probability of traversing an attack graph along a certain path is typically computed as a function of some measure associated with each vulnerability. Allodi and Etalle [7] theorized that tailored attacks are carried conducted against a specific target or a specific set of targets. These type of attacks require some level of engineering or technical sophistication in addition to deployment of the attack by the attacker. In prior work, they concluded the vast majority of attacks in the wild are enabled by five to ten exploits [9], and the refresh time of attacks in the wild is as slow at 600 days (i.e., the same exploits are re-used in the wild for almost two years before they are substituted at scale with a new attack) [10]. With this background knowledge, Allodi and Etalle simulated a series of attack scenarios against an Industrial Control System to determine the probability of success and upfront costs (i.e., resources) an adversary must invest to obtain their desired outcome (e.g., ransomware, website defacement). Their focus on attack characteristics and the attacker environment is similar but different from the approach we intend to address RQ2. While their attacker is a notional one, we intend a data driven approach using identified threat groups found in the MITRE ATT&CK matrix.

3.4 ADDRESSING THE RESEARCH GAP

To be effective, security teams need to understand vulnerabilities in the context of business risk, and then use that data to prioritize their remediation while exerting the least amount of effort. Prior research has demonstrated the ability to examine adversary capabilities, vulnerability management, and exploit prediction at a particular point in time or with isolated threat scenarios. However, little research has been done to create an end-to-end prioritization approach which encompasses the entire vulnerability management life cycle. The automated and flexible graph-native approach we propose addresses this gap by:

- Extracting dozens of essential features about the vulnerability, including its potential for harm, the degree to which it is exploitable, and how frequently the vulnerability is targeted by adversaries (relates to RQ1).
- Leveraging the ability of property graphs to offer a flexible schema where you can

constantly add and drop attributes to extend or shrink your data model, create hierarchies with different levels of granularity, and combine multiple dimensions to better manage big data (relates to RQ1).

- Performing an assessment of current and predicted future attacker activity based on known tactics and techniques (relates to RQ2).
- Correlating threat and exploit intelligence from publicly available authoritative sources (relates to RQ1).
- Devising an approach to convert raw data about threat indicators into contextual risk scores (relates to RQ2).
- Identifying how important the affected asset is to an organization in any industry (relates to RQ2).
- Inferring indirect facts and hidden relationships which can further inform our results (relates to RQ1).

As might be expected, parsing through the open source cyber threat intelligence data we defined cannot be reasonably accomplished by a human analyst, therefore, automating its correlation and analysis using graph native algorithms is absolutely essential. We can also leverage the Application Programming Interfaces (APIs) and data feeds maintained by NIST to provide awareness of the changing threat landscape while allowing for dynamic and continuous assessment of the underlying network architecture. The research we propose will provide benefits to organizations seeking to create high-level strategies to examine cybersecurity posture in a manner that is predictive not just reactive.

3.5 CHAPTER SUMMARY

In this chapter, we considered research efforts in creating cybersecurity ontologies, vulnerability categorization, and vulnerability prioritization that are similar, different, and inform this work. We began by exploring approaches for creating a cyber ontology suitable for determining the relationship, if any, between individual vulnerabilities published in the NVD. We showed how data sets managed by NIST and MITRE are reliable, authoritative sources for vulnerability reporting but still require data enrichment from ancillary sources. Most notably, we examined how the temporal and environmental scoring algorithm could be extended to provide context that is specific to the organization. We also noted how prior

research contends with limitations, incompleteness, and known biases in any of the public databases (Chapter 2) used to characterize vulnerabilities and exploits. These databases include only the vulnerabilities and exploits that are publicly reported and known to the security community. Next, we described state-of-the-art techniques for ranking vulnerabilities to determine their likelihood of exploit based on past observations by reputable sources in the white-hat community (e.g., ExploitDB, attack signatures) and musings gleaned from forums on the dark web. Finally, we concluded by showing that adversarial attacks against a network can be computationally intensive to represent using logical attack graphs and presented an alternative for more efficiently capturing the properties of both the network configuration and the data sets related to the vulnerability.

Chapter 4

ESTABLISHING THE CYBERSECURITY ONTOLOGY

In this chapter, we present the work conducted to address RQ1:

RQ1: What are the factors that can be used to model attack vectors and security threats based on the skill level of a cyber adversary and their motivation to target a specific industry domain (e.g., national defense, higher education, finance, health care)?

We captured different sources of cyber threat intelligence that we analyzed to compile a list of prioritized recommendations. The correlation of data sources and their relationship to the software vulnerability lifecycle discussed in Section 2.1.3 is summarized in Figure 47. Each activity in the lifecycle (i.e., middle row) can be directly mapped to a NIST or MITRE data source (i.e., bottom row) needed for analysis. The exception is "Targets a Web Site" which represents the weakness or misconfiguration that allows an attacker to gain some level of control of the site.

In this section, we describe methods used to collect, parse, and store the data sets which in some cases we filtered prior to analysis to gain a better understanding of what they hold. The vulnerability data we examined consists of basic information and high-dimensional attributes for the observed data noted in Sections 4.1 to 4.6. The web site and data feeds used to access each source is shown in Table 5. Using data provided by other sources has been proven useful for enriching a CVE-ID with ancillary information. The majority of the cyber intelligence data is available in a structured format (e.g., XML, JSON, XLSX). However, it can be massive, noisy, and inconsistent, which can exacerbate the already challenging problem of feature selection [150]. Therefore, we collected as many fields as possible from each data feed in anticipation of how it might eventually inform our ranking algorithm. The Exploit database required screen scraping since a structured data feed was not available.

A variety of candidate data sources were previously defined in Chapter 2. In order to compile the required data sets, these sources were studied and evaluated to ensure the data obtained would be appropriate, relevant and of sufficient high quality. The main requirement for the selection of each specific data source was that the data it provides must be publicly available and primarily in the form of structured data sets (e.g., JSON, XML, CSV). The information provided by a data source was also evaluated in terms of accuracy, consistency

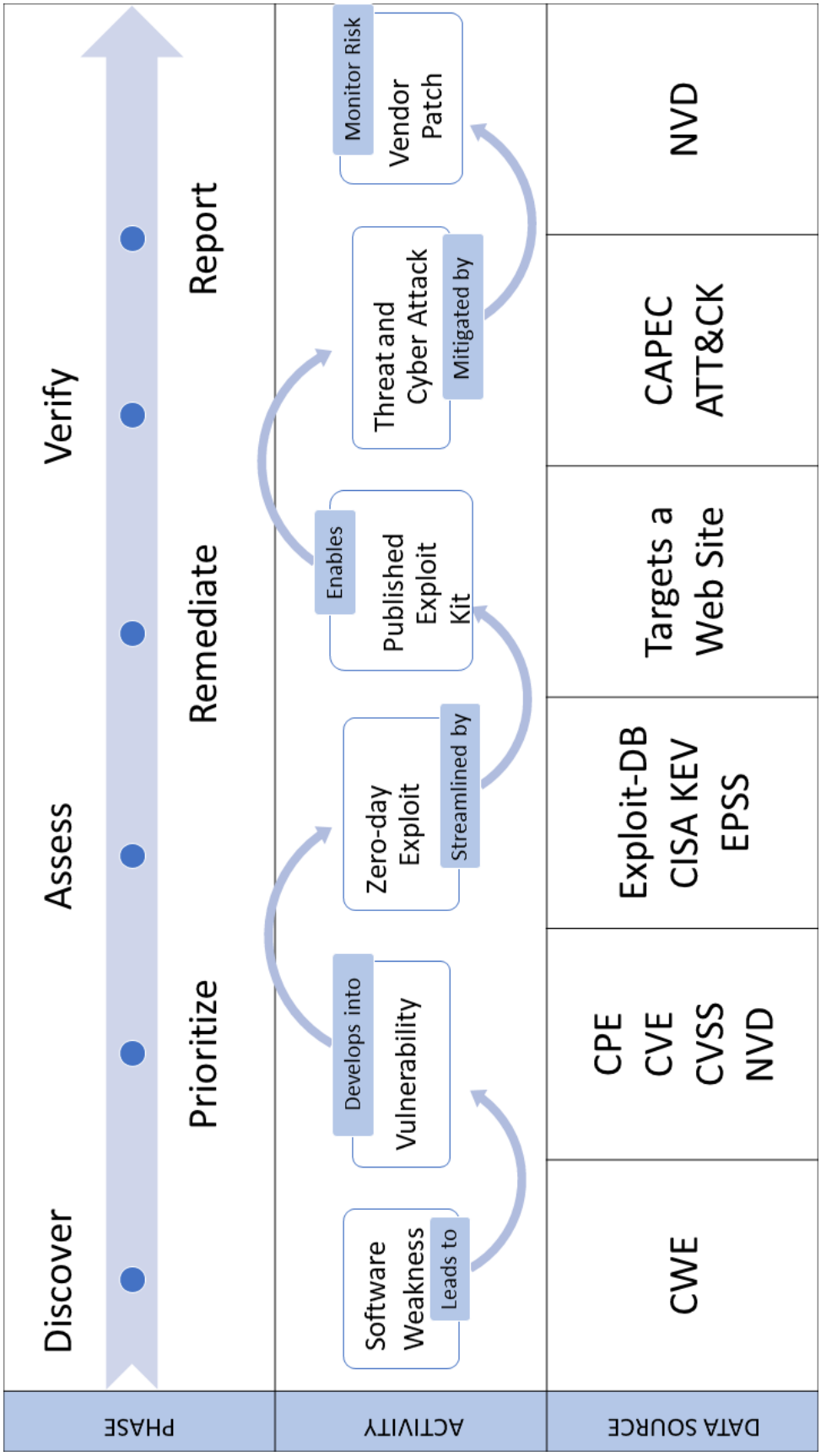


Figure 47: Software vulnerability lifecycle phases as viewed in relation to our proposed cybersecurity ontology.

Table 5: Summary of collection methods for cyber intelligence data sources.

Dictionary	**Description**
ATT&CK	ATT&CK tactics, techniques, and group descriptions.
	Downloaded from `attack.mitre.org/resources/working-with-attack/`
CAPEC	CAPEC ID to attack pattern.
	Downloaded from `capec.mitre.org`
CPE	CPE list to product and vendor information.
	Downloaded from `nvd.nist.gov/products/cpe`
CWE	CWE IDs to weakness description.
	Downloaded from `cwe.mitre.com`
NVD	CVE-ID to vulnerability.
	Downloaded from `nvd.nist.gov/vuln/data-feeds`
Exploit Database	CVE-ID to proof of concept code.
	Screen scrapped from `cve.mitre.org/data/refs/refmap/source-EXPLOIT-DB.html`
CISA KEV	CVE-ID to date added as exploit.
	Downloaded from `www.cisa.gov/known-exploited-vulnerabilities-catalog`
EPSS	CVE-ID to probability of exploit score and percentile.
	Downloaded from `www.first.org/epss/data_stats`

and completeness by considering additional external references from other well-established sources or standards. The ease of data extraction from a given source was an additional requirement that was taken into consideration. The data sources were combined in order to produce an initial, rich dataset with a variety of features (or dimensions). All data was collected as of December 31, 2021.

4.1 SOFTWARE WEAKNESSES DATASET

A vulnerability in the NVD can be associated with one or more software or hardware weaknesses using the CWE ID. The associated hierarchy in Figure 48 shows the categories of information for which we can analyze an CWE entry. For example, related weaknesses will allow us to ascertain a parent-child and peer relationship between CWE IDs. We may be able to distill the CWE into macro categories (i.e., smaller parent nodes) to reduce noise and help refine our ranking model. We can both drill up and drill down to determine which approach produces the best results.

The NVD integrates CWE into the scoring of vulnerabilities by incorporating a cross section of the overall CWE structure. CWE View-1003 (Figure 49) contains "Weaknesses for Simplified Mapping of Published Vulnerabilities". A view is just another way to categorize or group CWE IDs. For our purposes, we restrict data collection to the CWE IDs associated with this view to closely align with the expected mappings from the NVD. View-1003 is currently software centric, so if cyber practitioners later decide to include hardware weaknesses, it will be necessary to collect the CWE View-1194 related sections.

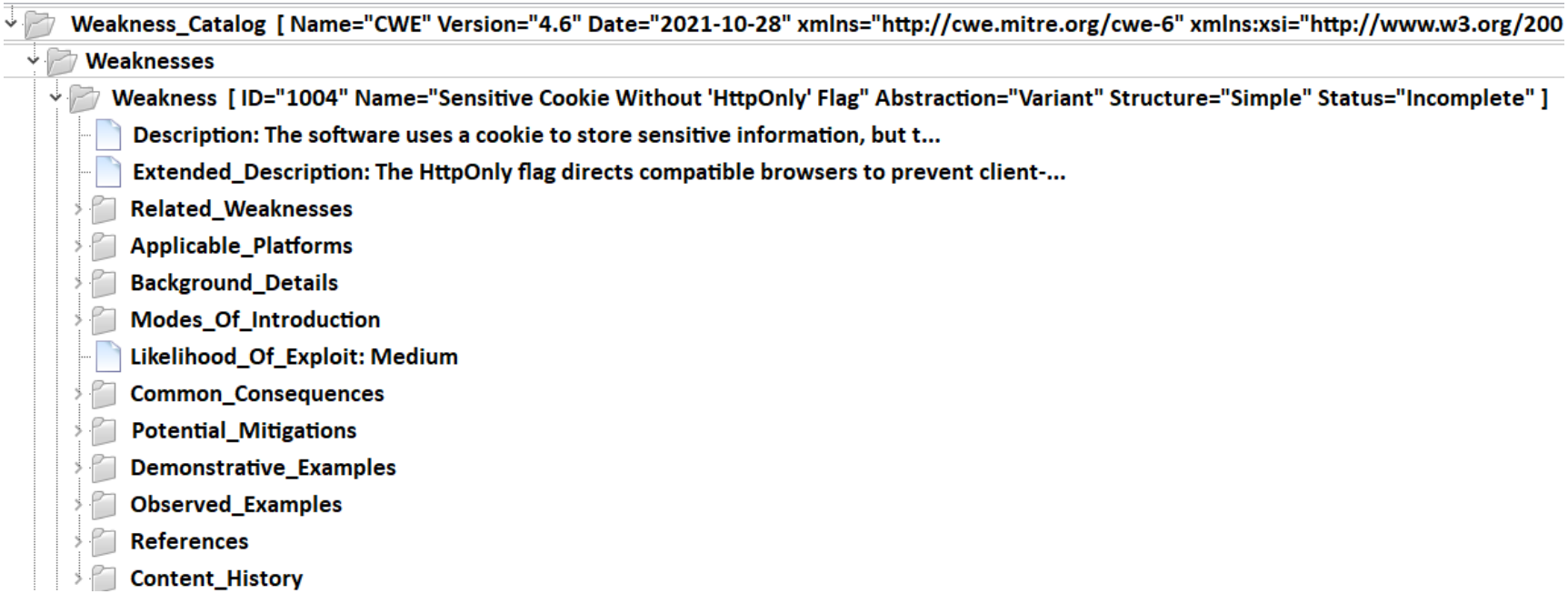


Figure 48: CWE element hierarchy (Reproduced from [100]).

.

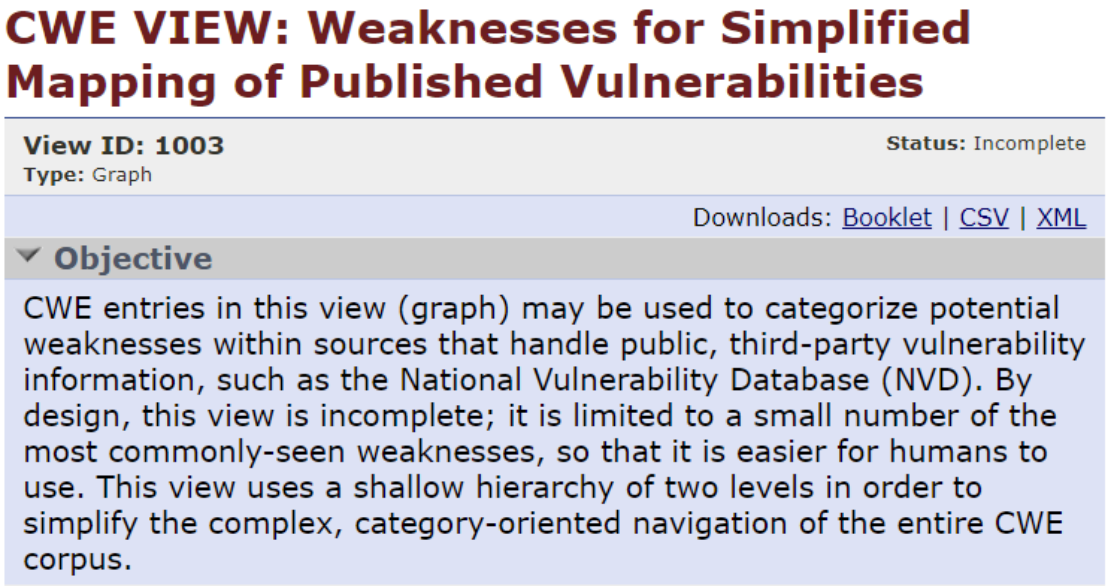


Figure 49: CWE View-1003 which maps to the NVD (Reproduced from [100]).

The metadata shown in Table 6 was extracted from the CWE data feed. In addition to the NVD relationship, the CWE data feed provides a link to CAPEC via the Related Attack Pattern(s) entry. This relationship will allow us to examine our ranking approach from the adversary's point of view.

While there are about 700 identified weaknesses in the CWE, there appear to be only eight different consequences or technical impacts which lead to failure. In other words, a weakness that an attacker successfully exploits will result in one of the eight technical impacts or consequences from that weakness. Within each CWE entry the consequence scope field lists the technical impacts that can result from each associated weakness in the CWE. We can potentially condense hundreds of types of errors into a smaller set of technical impacts which may simplify our ranking approach. The technical impacts of software weaknesses are:

- Read data
- Modify data
- Denial-of-Service: unreliable execution
- Denial-of-Service: resource consumption

Table 6: CWE feature extraction.

Field	Example
Weakness ID	CWE-400
Name	Uncontrolled Resource Consumption
Related Weaknesses	ChildOf CWE-664
Consequence Scope	Access Control
Technical Impact	DoS: Crash, Exit, or Restart
Detection Method	Automated Static Analysis
Likelihood of Exploit	High
Potential Mitigations	Design throttling mechanisms into the system architecture.
Phase	Architecture and Design
Related Attack Pattern(s)	CAPEC-ID 147

- Execute unauthorized code or commands
- Gain privileges / assume identity
- Bypass protection mechanism
- Hide activities

The CWE dataset consists of 923 entries. There were 23 additional entries marked as deprecated. We standardized the CWE naming convention by pre-pending 'CWE-' to each numeric entry. We also manually added two CWE records which are referenced in the NVD, but are not actually CWE entries. The manual entries, which were typically noted during the initial phases of vulnerability analysis, were:

- NVD-CWE-noinfo, Insufficient Information
- NVD-CWE-Other, Other

4.2 VULNERABILITY DATASET

This study covers vulnerabilities published between January 1, 2019 through December 31, 2021. The vulnerabilities were regularly collected and hosted in the compiled data set

XML Schema Version 1.1 : NVD JSON 1.1 Schema

Feed	Updated	Download	Size (MB)
CVE-Modified	11/25/2022; 8:00:02 PM -0500	META	
		GZ	0.33 MB
		ZIP	0.34 MB
CVE-Recent	11/25/2022; 8:00:00 PM -0500	META	
		GZ	0.09 MB
		ZIP	0.09 MB
CVE-2022	11/25/2022; 3:00:12 AM -0500	META	
		GZ	4.23 MB
		ZIP	4.23 MB

Figure 50: NVD JSON data feed update schedule (Reproduced from [114]).

until the cut-off date of December 31, 2021. Our snapshot of the NVD reflects that date range. There are likely to be more vulnerabilities included in the period under examination today. This is due to the lag of vulnerabilities officially receiving a CVE-ID and entering the system.

The list of CVE-IDs was collected using the NIST-provided JSON data feeds. Each vulnerability in the file includes a description and associated reference links from the CVE dictionary feed, as well as CVSS base scores, vulnerable product configurations, and weakness categorizations. The available data feeds are shown Figure 50. According to the NVD website,[1] the "year" feeds (e.g., CVE-2021), where the year represents the publication year, are updated once per day, while the "CVE-Recent" and "CVE-Modified" feeds are updated every two hours.

In order to further contextualize the vulnerability entries, several related dictionaries were also downloaded (Table 5). These files were used through mapping and lookup functions to translate the applicable numerical values and IDs into more meaningful information, as needed. The metadata shown in Table 7 was extracted from the NVD data feed. Each CVE-ID can be associated with one or more vendor products (CPE ID) and software weaknesses (CWE ID).

[1] `https://nvd.nist.gov/vuln/data-feeds`

Table 7: NVD feature extraction.

Field	**Example**
CVE-ID	CVE-2018-6156
CPE-Affected	cpe:2.3:a:google:chrome:*:*:*:*:*:*:*:*
Vendor	google
Product	chrome
Version	ANY
CVE Description	Incorrect derivation of a packet length in WebRTC in Google Chrome prior to 68.0.3440.75 allowed a remote attacker to potentially exploit heap corruption via a crafted video file.
CVSS Base Score	8.8
CVSS Vector	CVSS:3.0/AV:N/AC:L/PR:N/UI:R/S:U/C:H/I:H/A:H
Exploitability	2.8
Impact	5.9
Modified Date	6/27/2019
Patch Reference(s)	https://chromereleases.googleblog.com/2018/07/stable-channel-update-for-desktop.html
Related CWE(s)	CWE-119
Severity	High

4.3 VENDOR PRODUCT DATASET

The Common Platform Enumeration (i.e., list of CPE IDs) was collected using the NIST provided data feeds. We collected the entire CPE V2.3 dictionary to ensure complete coverage for vendor products associated with any vulnerabilities we might encounter in the NVD. As shown in Figure 51, each entry in the XML file defines a specific hardware device, operating system, or application software.

Figure 51: CPE XML element entry (Reproduced from [29]).

Table 8: CPE feature extraction.

Field	**Example**
CPE-ID	cpe:2.3:a:harman:adobe_air_sdk:33.0.0.168:*:*:*:*:*:*:*
Title	Harman Adobe AIR SDK 33.0.0.168
Part	a
Vendor	harman
Product	adobe_air_sdk
Version	33
Update	0
SwEdition	0
Target_SW	168
Target_HW	*
Language	en-US
Reference(s)	`https://airsdk.harman.com/release_notes`

The CPE dictionary is a large XML catalog that contains entries formatted using version 2.2 and the newer version 2.3. We will need to separate these entries to ensure compatibility with CVSSV3 base scores in the NVD which use CPE V2.3 for its entries. The metadata shown in Table 8 was extracted from the CPE data feed using the CPE formatted string notation defined in Section 2.1.

Entries marked as deprecated, as shown in Figure 52, were excluded. A deprecated CPE ID is one that previously appeared in the Official CPE dictionary but has since been replaced by one or more other CPE IDs. CPE IDs are deprecated for various reasons, such as when the original CPE name is discovered to be incorrect, when a more specific CPE name is added, and when a vendor name or product name evolves.

The language component indicates a language-specific release of a product (e.g., English, Spanish, Japanese). We will restrict the CPE IDs of interest to those specific to US English. Entries that apply to other languages were ignored. We applied these filtering constraints to the entire published CPE dictionary. The CPE dictionary currently contains more than 15,000 CPE entries representing more than 3,000 products from approximately 200 vendors.

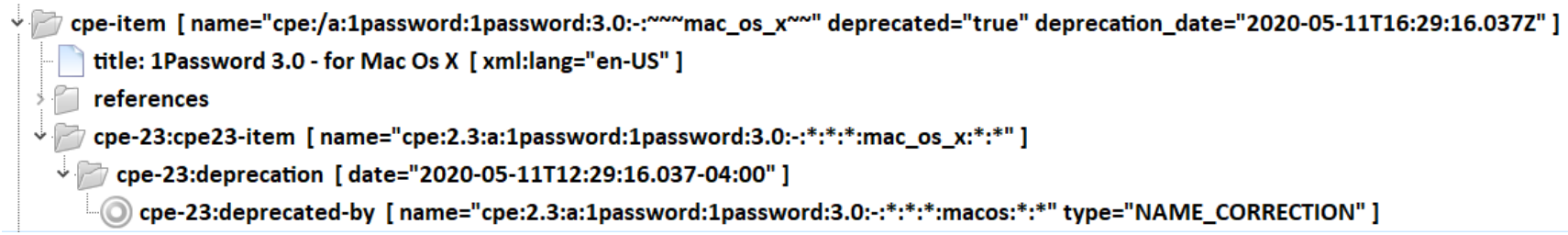


Figure 52: Deprecated CPE element (Reproduced from [29]).

4.4 ATTACK PATTERN DATASET

CAPEC is focused on application security and describes the common attributes and techniques employed by adversaries to exploit known weaknesses. The attack patterns found in CAPEC provide a linkage to both the CWE and MITRE ATT&CK matrices. We noted there is not necessarily a 1:1 relationship between all elements in either set. The associated hierarchy is shown in Figure 53. For example, related attack patterns will allow us to ascertain a parent-child and peer relationship between attack pattern IDs.

CAPEC currently contains 541 attack patterns. The metadata shown in Table 9 was extracted from the CAPEC data feed.

Table 9: CAPEC feature extraction.

Field	**Example**
Attack Pattern ID	CAPEC-1
Name	Accessing Functionality Not Properly Constrained by ACLs
Likelihood of Attack	High
Typical Severity	High
Likelihood of Exploit	High
Related Attack Patterns	ChildOf CAPEC-122
Skills Required / Skill Level	Low
Consequence Scope	Confidentiality
Mitigation(s)	In a J2EE setting, administrators can associate a role
Related Weakness(es)	CWE-ID 276
Taxonomy Mapping	ATTACK
Entry Id	1574.010
Entry Name	Hijack Execution Flow: Services File Permissions Weakness

Figure 53: CAPEC element hierarchy (Reproduced from [18]).

Figure 54: CAPEC domains of attack (Reproduced from [18]).

Another way to analyze CAPEC is through the Domains of Attack shown in Figure 54. The tree relationship is used group attack patterns based on an abstract characterization of the methodologies or techniques employed. For example, the attack patterns grouped under the Supply Chain category focus on the disruption of the supply chain lifecycle by manipulating computer system hardware, software, or services for the purpose of espionage, theft of critical data or technology, or the disruption of mission-critical operations or infrastructure [18]. CAPEC attack patterns, through their defined mitigations, can also support our goal to identify relevant remediations for weaknesses uncovered through successful attacks. We may also be able to model the attacker in terms of skill and expertise.

We collected 545 unique instances of CAPEC identifiers. The associated CAPEC-Taxonomy contains 162 unique entries that provide linkages to MITRE ATT&CK techniques. As shown in Table 10, we standardized the technique_id by pre-pending 'T' to match the ATT&CK Enterprise techniques naming conventions.

Table 10: Example of post-processing of CAPEC taxonomy to provide standardization with MITRE ATT&CK techniques.

CAPEC Technique ID	**MITRE Technique ID**	**CAPEC Post Processing**
1498.002	T1498.002	T1498.002

MITRE ATT&CK describes 14 Attack Enterprise Tactics, 188 Attack Enterprise Techniques, and 378 sub-techniques. We noted that sub-techniques T1547.001 and T1553.003 have descriptions that were too long based on limitations of our database import (i.e., greater than 4000 chars). The descriptions were truncated from the end and subsequently imported successfully with no loss of context. Several inconsistencies were noted when completing the linkage between MITRE ATT&CK techniques and CAPEC. Specifically, we identified instances where one data source does not share the reciprocal information shown in another.

- CAPEC-66[2] does not reference an association with Technique ID T1017 Application Deployment Software. However this association is included in the CAPEC comprehensive dictionary CSV download file.
- CAPEC-657 has a known association with Technique ID T1017 Application Deployment Software. However, the web link[3] in the taxonomy entry ID fields navigates instead to T1072 Software Deployment Tools.[4]

4.5 EXPLOIT DATASET

MITRE publishes a web page, updated daily, that provides a mapping between the CVE and ExploitDB. Since there is no structured data feed, we wrote scripts to scrape and parse the data. It is a one-to-many mapping of an identified exploit kit to the vulnerabilities which are the target of that exploit. The metadata shown in Table 11 was extracted from the web page.

[2] `https://capec.mitre.org/data/definitions/66.html`
[3] `https://attack.mitre.org/wiki/Technique/T1017`
[4] `https://attack.mitre.org/techniques/T1072/`

Table 11: ExploitDB feature extraction.

Field	**Example**
Exploit ID	EXPLOIT-DB-10102
CVE-ID(s)	CVE-2009-4186

The CISA KEV provides real-time updates via email alerts when a newly, identified CVE-ID is exploited. All known exploits to date are available via a CSV download from which we extracted the CVE-ID and date added to the catalog. In Table 12, we note a disproportionate number of CVE-IDs were added on November 3, 2021. This is likely due to a bootstrap to initialize the catalog when it was established as noted in Section 2.3.2.

Table 12: CVE-IDs reported in CISA exploit catalog by date added.

Date Added	**Num. CVE-IDs**
11-03-2021	287
11-17-2021	4
12-01-2021	5
12-10-2021	13
12-15-2021	2

The EPSS, Section 2.3.3, provides an API[5] that allows users to query for the most recent CVEs or CVEs for a particular. To maintain the same point of reference for all data collected, we used the CSV download to obtain more than 80,000 entries related to CVE-IDs in our data set. The metadata shown in Table 13 was extracted from the CSV file.

[5] `https://api.first.org/data/v1/epss`

Table 13: EPSS feature extraction.

Field	Example	Note
CVE	CVE-2019-2725	The CVE identifier as specified by MITRE's CVE List
EPSS	0.962880000	The EPSS score representing the probability [0-1] of exploitation in the wild in the next 30 days (following score publication)
Percentile	0.999960000	The percentile of the current score, the proportion of all scored vulnerabilities with the same or a lower EPSS score

4.6 ADVERSARY TACTICS AND TECHNIQUES DATASET

The MITRE ATT&CK matrices are focused on network defense and describe the operational phases in an adversary's lifecycle and details the specific tactics, techniques, and procedures (TTPs) that Advanced Persistent Threat (APT) groups use to execute their objectives while targeting, compromising, and operating inside a network. Many attack patterns enumerated by CAPEC are employed by adversaries through specific techniques described by ATT&CK. This enables contextual understanding of the attack patterns within an adversary's operational lifecycle.

MITRE provides an Excel spreadsheet representation of the ATT&CK dataset that includes a master spreadsheet containing all object types, with individual spreadsheets for each object type. The individual type spreadsheets break out relationships (e.g., procedure examples connecting groups to techniques) into separate sheets by relationship type, while the master spreadsheet includes all relationship types in a single sheet. The metadata shown in Table 14 was extracted from the spreadsheets.

Table 14: MITRE ATT&CK feature extraction.

Field	**Example**	**Note**
Technique ID	T1548	
Technique Name	Abuse Elevation Control Mechanism	
Subtechnique(s)	T1548.002	
CAPEC ID	CAPEC-478	Shown on the MITRE web site, but not included in the spreadsheet
Tactic(s)	Defense Evasion	Convert text to a Tactic ID
Tactic ID	TA0009	
Platform(s)	Linux	
Software	3PARA RAT	
Type	malware	
Software Description	3PARA RAT is a remote access tool (RAT) programmed in C++ that has been used by Putter Panda (`https://attack.mitre.org/groups/G0024`).	Extract the threat group using the embedded URL (e.g., Putter Panda)
Used by technique(s)	T1110	
Group ID	G0024	
Group Name	Putter Panda	
Group Description	Putter Panda is a Chinese threat group that has been attributed to Unit 61486 of the 12th Bureau of the PLA's 3rd General Staff Department.	Extract the operating region (e.g., China)
Associated Group(s)	APT2, MSUpdater	Convert text to Group ID
Techniques Used	T1562.001	
Software Used	3PARA RAT	
Mitigation ID	M1036	
Mitigation Name	Account Use Policies	

CAPEC attack patterns and related ATT&CK techniques are cross referenced when appropriate between the two efforts. Specifically, CAPEC has identified View-ID 658 as the slice which covers patterns with a direct mapping to the ATT&CK matrices. This view includes 112 of 541 total entries in the CAPEC dictionary. In our research, the combination of MITRE ATT&CK and CAPEC may prove beneficial for prioritizing vulnerabilities which are exploitable by the highest number of threat agents.

4.7 CHAPTER SUMMARY

In Section 4.1 to 4.6, we described and analyzed the datasets that we used such as the NVD, CPE, CWE, CAPEC, Exploit DB, CISA, and MITRE ATT&CK. Using both structured data feeds and screen scraping, we identified a core set of features which provided input to our vulnerability ranking framework. We also determined a linkage through which each data set can be connected to create the knowledge graph schema we leveraged to facilitate the graph-based analysis and vulnerability model we implemented. As many of the sources had feature overlaps, it was possible to create a superset contextualizing each vulnerability further. For example, the CVE-ID can be used to link information between the software weaknesses (CWE), number of exploits (ExploitDB, CISA) which might be related to the vulnerability. The CAPEC ID that is included in both the NVD and MITRE ATT&CK matrices allows for a direct mapping of a CVE-ID to different attack techniques and tactics. Our goal while creating the ontology was to gather as many features as possible in order to identify variables which could potentially be used to personalize a mitigation approach.

Chapter 5

LINKING VULNERABILITIES TO THREAT ACTORS

In the previous chapter, we described the cyber intelligence datasets we used, such as the NVD, CPE, CWE, CAPEC, CISA KEV, and MITRE ATT&CK. In this chapter, we provide the results of our approach to develop the data-centric portions of the ranking framework, which includes combining the datasets to formulate a complete knowledge graph suitable for relevance model development, determining approaches to filter candidate vulnerabilities, and adding features to select and rank vulnerabilities for mitigation. For each step, we identify the dataset features we will use, the methods to accomplish the step, the challenges, and the methods to test and evaluate the quality of the outcome.

To address the research questions, we created several knowledge graphs using combinations of the cyber intelligence data sources defined in Chapter 4. This research starts with a data collection phase in which we gathered and normalized information used to define the analytical framework for our relevance ranking model. We are especially interested in data points that characterize techniques employed by cyber adversaries and their relationship to known vulnerabilities published in the NVD. After that, we will start to interrogate our framework to identify the features needed to create our ranking model. With the collected information, RQ1 and RQ2 can be addressed. We will then have a deeper understanding of the data relationships and how they can be leveraged to create a set of vulnerability ranking policies. In their work describing vocabularies for characterizing threat actors and related ontologies, Mavroeidis and Hohimer [101] indicated:

> *The ATT&CK Groups knowledge base lacks proper structurality and relationships between adversaries and their targets and between adversaries and their motivations. Information such as targeted countries and sectors and threat group motivations is embedded within the general description of a group and can be unstructurally searched using the ATT&CK portal. However, the vocabularies utilized to specify a group's targets and their motivations are not available, limiting searchability, and consequently, the ability to extract more relevant information.*

Here, we describe our approach to create the necessary standard vocabulary using publicly available data sources that can be linked to existing cyber intelligence (Chapter 4).

5.1 CONSTRUCTING A KNOWLEDGE GRAPH

We developed a knowledge graph that maps the attack relationships through an organization's software architecture using Neo4j [111], a NoSQL database optimized for graphs. Figure 55 shows the components of our system. We developed XML, CSV, and JSON parsers to ingest data from various cyber intelligence sources described in Chapter 4 and mapped them to a common relational database model. We analyzed elements of the database model for potential attack scenarios, nodes, relationships, and associate them with a notional software architecture. We stored the resulting knowledge graph in a our graph database. Once we stored technical metadata in the knowledge graph and created relationships between them, we had a semantic view of the data environment needed to pursue further analysis of our ranking models.

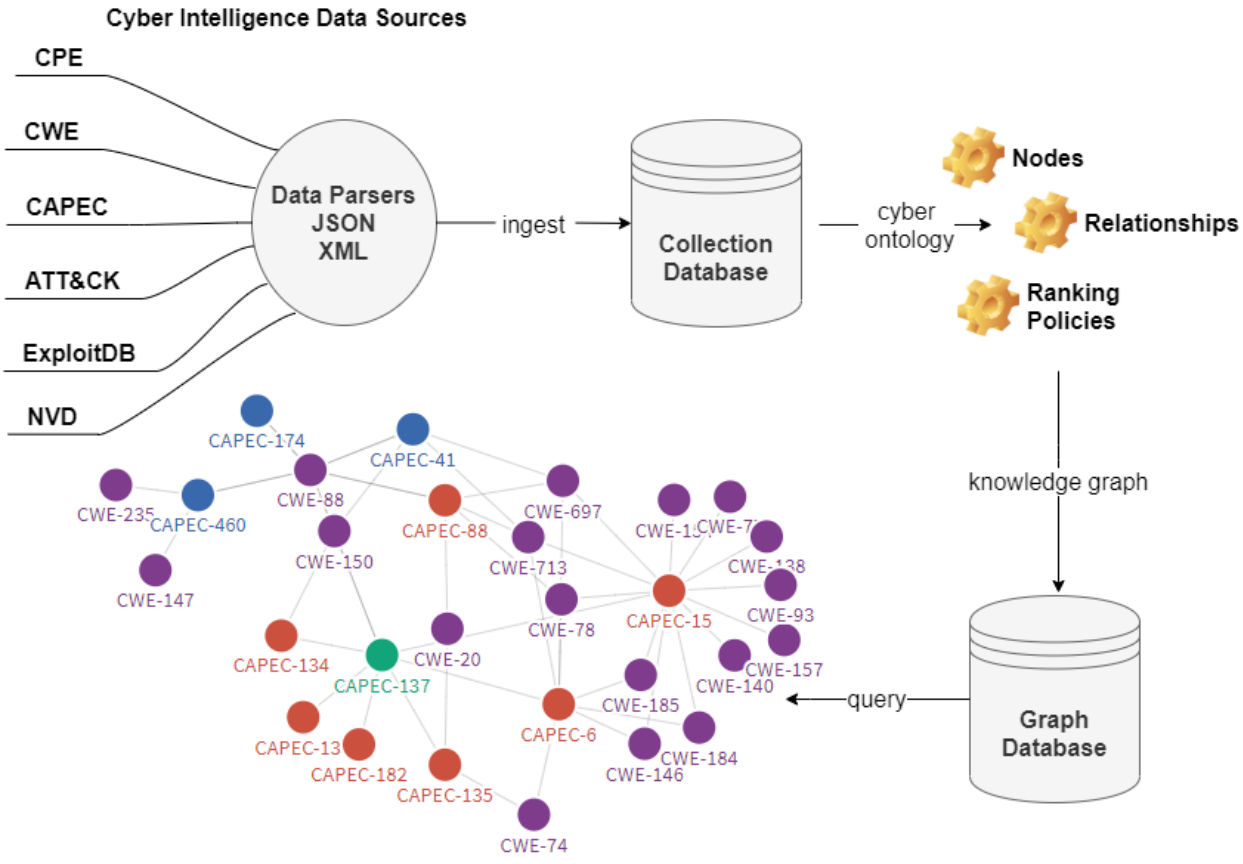


Figure 55: Components of graph knowledge base.

The knowledge graph is used for both storing the data and performing analysis to support our ranking policies. The overall schema of the knowledge graph is presented in Figure 56. The graph represents a network of real-world entities, in this study the cyber intelligence discussed in Chapter 2, and illustrates the relationship between them. The legend which describes the node labels and relationships is shown in Table 15. To achieve rapid analysis, it is critical to use automation and integration to collect and process large volumes of data from many sources. Most organizations lack the resources and inclination to continuously perform this type of aggregation. In the following sections, we will describe the process of building the knowledge graph that will allow us to link attackers to vulnerabilities.

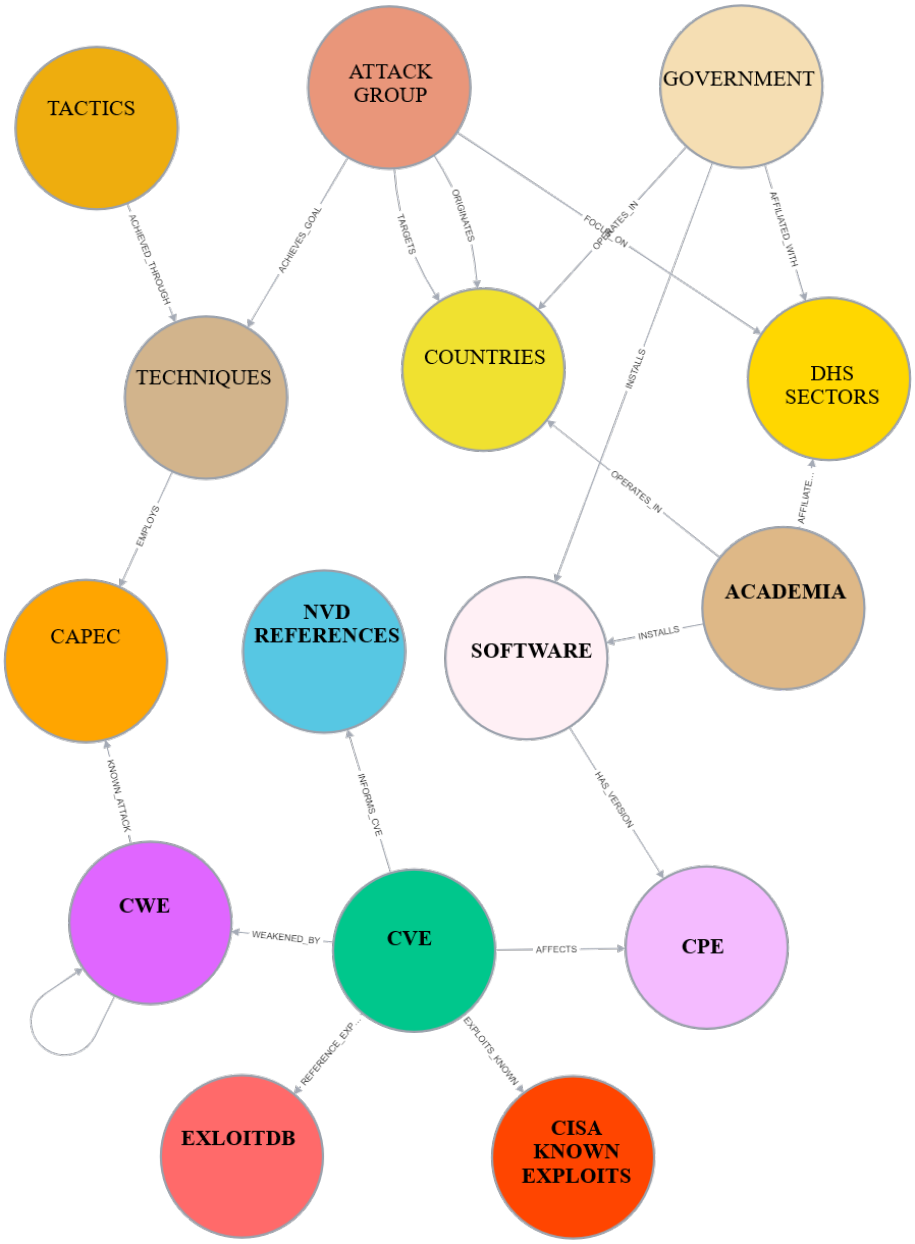


Figure 56: Graph schema representing the entities of the knowledge graph and the relationship between them.

Table 15: Legend for node labels and relationships in knowledge graph schema.

Label	**HEX Color**	**Relationship**	**Label**	**HEX Color**
NVD CVE	Caribbean Green	REFERENCE EXPLOIT	ExploitDB	Indian Red
NVD CVE	Caribbean Green	EXPLOITS KNOWN	CISA Exploit Catalog	Red Orange
NVD CVE	Caribbean Green	WEAKENED BY	CWE	Medium Orchid
CWE	Medium Orchid	KNOWN ATTACK	CAPEC	Orange
CAPEC	Orange	EMPLOYS	Attack Enterprise Techniques	Tan
Attack Groups	Dark Salmon	ACHIEVES GOAL	Attack Enterprise Techniques	Tan
Attack Groups	Dark Salmon	ORIGINATES	Countries	Dandelion
Attack Groups	Dark Salmon	TARGETS	Countries	Dandelion
Attack Groups	Dark Salmon	FOCUS ON	DHS Sectors	Golden
Attack Enterprise Tactics	Dark Tangerine	ACHIEVED THROUGH	Attack Enterprise Techniques	Tan
NVD CVE	Caribbean Green	AFFECTS	CPE	Mauve
DHS Sectors	Golden	AFFILIATED WITH	Organizations	Grayish Blue
Organizations	Grayish Blue	OPERATES IN	Countries	Dandelion
Organizations	Grayish Blue	INSTALLS	Software	Lavender Blush
Software	Lavender Blush	HAS VERSION	CPE	Mauve
NVD CVE	Caribbean Green	INFORMS	NVD References	Malibu

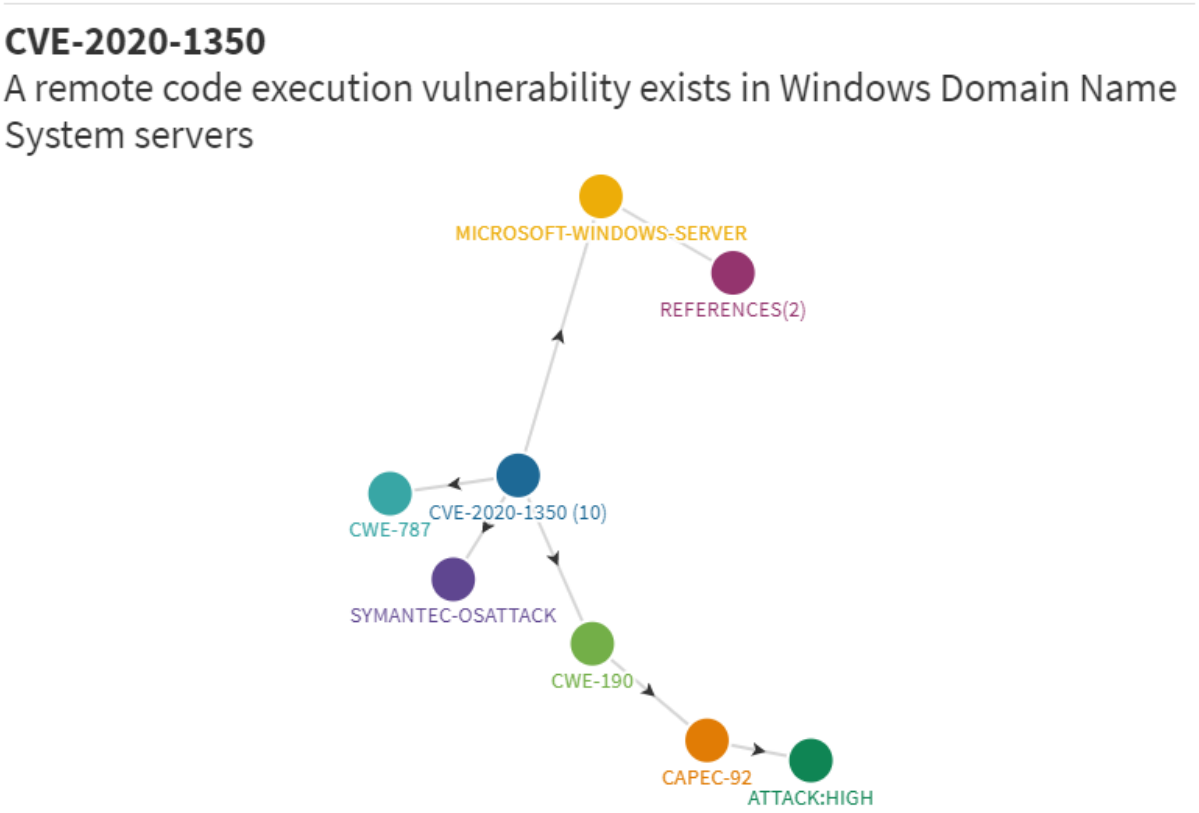


Figure 57: SigRed CVE-2020-1350 with severity 10 (critical) affects nine software products made by Microsoft.

For best results when creating a knowledge graph, we need to have a pre-formulated set of access patterns (i.e., analytical questions) to ensure the graph schema can answer those questions without a significant amount of refactoring or degraded performance. The initial set of questions we will use to formulate the ranking model are described in Section 5.2. Another thing we can do is to generate several models employing different sets of variables designed to answer different types of questions. Figures 57 and 58 provide small-scale examples of how we aggregated our cyber intelligence data to begin analysis of individual CVE-IDs. In Figure 57, the inherent weaknesses (CWE-190, CWE-787) have no known threat groups. The likelihood of attack is high (CAPEC-92). A known operating system attack has been reported by Symantec. There are two known mitigations (i.e., patch references in the NVD).

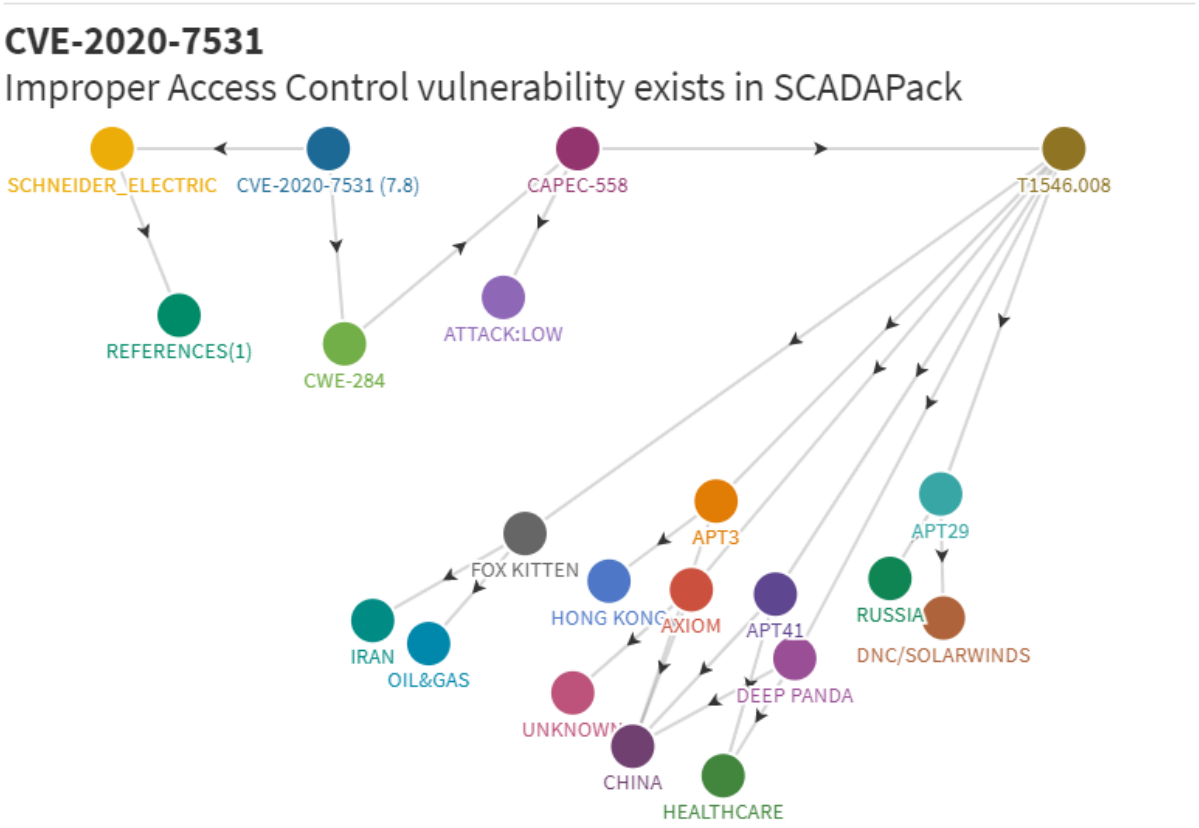


Figure 58: CVE-2020-7531 with severity 7.8 (high) affects one software product made by Schneider Electric.

In Figure 58, the inherent weakness (CWE-284) is targeted by six threat groups (APT29, APT3, APT41, AXIOM, Deep Panda, and Fox Kitten) who use the same tactic (T1546.008). The likelihood of attack is low (CAPEC-558). There is one known mitigation (i.e., patch references in the NVD).

Our goal is to develop a methodology focused on disrupting adversaries more likely to target an organization based its operating sector and where it is geographically located. The ability to create a data-driven, threat-centric model to determine which CVEs might be targeted is hampered by disparities and gaps in traditional vulnerability data sources. You need multiple sources of cyber intelligence to form a complete picture of potential and actual threats. While the majority of the data presented in Chapter 4 is collected and maintained by the same or cooperating research groups (e.g., FIRST, MITRE, DHS), the

data itself is not captured in a standard way that allows it to be easily correlated. As a part of this research, we further expanded RQ1 to determine how a plethora of data feeds can be logically linked in a way that we can associate a threat actor (i.e., APT group) to a set of vulnerabilities which match the tactics and techniques typically used by the group. The vulnerability recommendations presented are powered by the knowledge graph that represents the underlying knowledge layer of our system. The relationships in this graph connect a large set of cyber threat intelligence and adversary information representing the threat model used to produce personalized rankings.

5.2 DEFINING A STANDARD SET OF SECTORS

Critical infrastructure (CI) can be a subject of interest when they are exposed to threats either due to natural disasters such as hurricanes or man-made threats such as cyber attacks. As depicted in Figure 59, DHS manages 16 critical infrastructure sectors "whose assets, systems, and networks, whether physical or virtual, are considered so vital to the United States that their incapacitation or destruction would have a debilitating effect on security, national economic security, national public health or safety, or any combination thereof" [41]. The operations of critical infrastructure are highly dependent on each other and interconnected. For example, a water supply is needed to cool natural gas lines and gas supplies are needed to fuel transportation and shipping [70]. As a result of these inter-dependencies and the potential impact on a nation's economy, critical infrastructure is at high risk for cyber threats and vulnerabilities that might be targeted by adversaries. In an effort to protect CI, Presidential Policy Directive 21 (PPD-21): *Critical Infrastructure Security and Resilience* established a national policy to strengthen and maintain secure, functioning, and resilient critical infrastructure [120]. PPD-21 acknowledges that as physical and cyber elements of critical infrastructure are inextricably linked, so are the vulnerabilities. In addition to sectors, DHS can also create subsectors. Currently, only the Government Facilities Sector has three designated subsectors. These include Elections, National Monuments and Icons, and Education Facilities [159]. In this research, the CI sectors and subsectors are used to provide an affiliation for both threat actors (Section 4.3) and the organizations they target (Section 5.2).

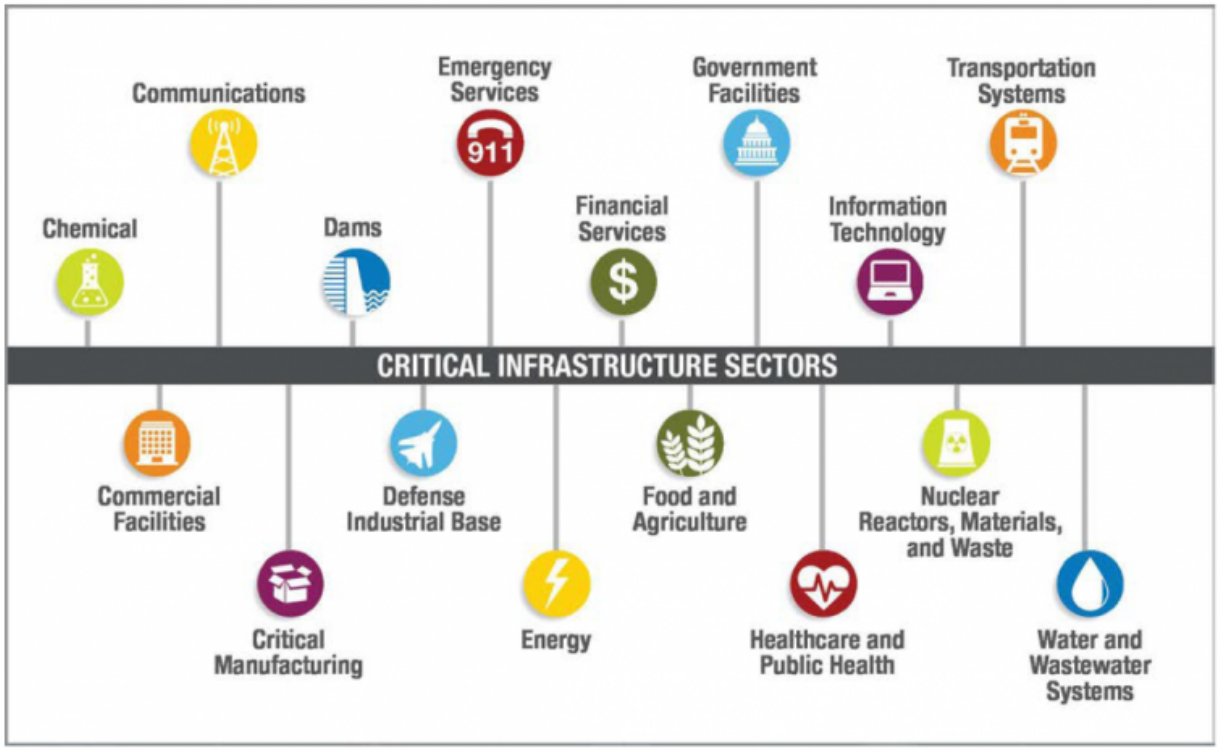


Figure 59: The DHS critical infrastructure sectors (Reproduced from [70]).

5.3 DEFINING STANDARD LOCATIONS

A standard nomenclature is needed to determine the country or region of origin for cyber threat actors and country of residence for organizations they target for attack. The U.S. State Department maintains a list of independent states that can be downloaded in a CSV format. In this list, the term "independent state" refers to a people politically organized into a sovereign state with a definite territory recognized as independent by the U.S. [158]. Table 16 contains an excerpt of the entire list of 262 countries.

Table 16: Excerpt from the State Department's list of independent states [158].

Short-form Name	**Long-form Name**	**Capital**
Afghanistan	Islamic State of Afghanistan	Kabul
Albania	Republic of Albania	Tirana
China	People's Republic of China	Beijing
Russia	Russian Federation	Moscow
United States	United States of America	Washington, DC

5.4 ASSIGNING ATTRIBUTES TO ADVERSARY GROUPS

Groups are sets of related intrusion activity that are tracked by a common name in the security community. Analysts track clusters of activities using various analytic methodologies and terms such as threat groups, activity groups, threat actors, intrusion sets, and campaigns. Some groups have multiple names associated with similar activities due to various organizations tracking similar activities by different names. Organizations' group definitions may partially overlap with groups designated by other organizations and may disagree on specific activity. Advanced Persistent Threat (APT) hacker groups have become a virtual extension of nation-states' military forces because of the potential damages and chaos caused by successful critical infrastructure cyber attacks. MITRE ATT&CK provides classifications of tactics and techniques that allow security teams to be very granular in describing and tracking adversarial behavior. In this research, we want to consider vulnerability ranking and prioritization from the view point of the attacker. MITRE provides a concise list of 129 threat groups [106] in their *Enterprise Framework* that can be associated with known techniques. Using their defined threat profiles, we further explored RQ1 to identify adversaries or threat groups who employ the same tactics and techniques.

5.4.1 WHERE ATTACKS ORIGINATE

First, we performed data mining on adversary groups descriptions, then used natural language processing to extract keywords to determine the country from which the group operates. For example, a *North Korean state-sponsored threat group* would be assigned to

North Korea with our mapping. Some descriptions, as shown in Table 17, include specific country names while other target broader regions (e.g., Middle East) or a continent (e.g., Asia, the Americas). In those non-specific instances, no country affiliation was assigned. We also mined the descriptions to determine year of origin (e.g., 2008) to ascertain the potential longevity of each group. If a year was not explicitly stated in the description, we used the creation date of the posting (e.g., has been active since at least 2009).

Each keyword phrase was assigned a country using the independent states previously identified in Section 5.3. We identified 77 of 128 country of origin affiliations, Table 18, for the attack groups as noted in the group description by extracting phrases similar to:

- **operating out of** Iran
- North Korean **state-sponsored threat group**
- **attributed to** China
- Chinese-**based threat group**

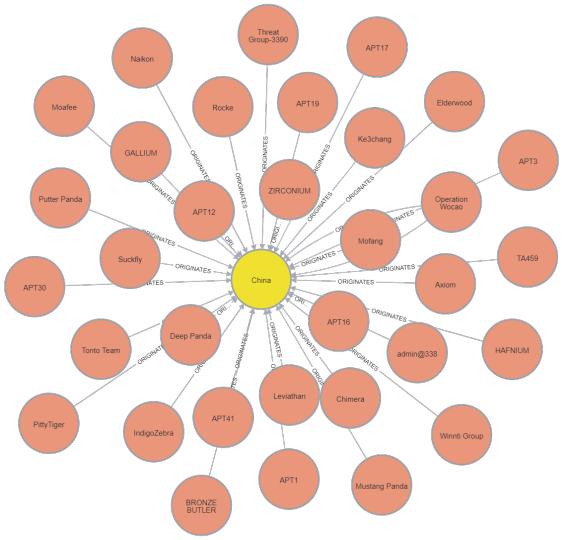
Table 17: Keywords in the description are used to assign an attack group to the country from which it operates.

ID	Name	Description
G0018	admin@338	admin338 is a **China-based** cyber threat group. It has previously used newsworthy events as lures to deliver malware and has primarily targeted organizations involved in financial, economic, and trade policy, typically using publicly available RATs such as PoisonIvy, as well as some non-public backdoors.
G0099	APT-C-36	APT-C-36 is a suspected **South America** espionage group that has been active since at least 2018. The group mainly targets Colombian government institutions as well as important corporations in the financial sector, petroleum industry, and professional manufacturing.
G1000	ALLANITE	ALLANITE is a suspected **Russian** cyber espionage group, that has primarily targeted the electric utility sector within the United States and United Kingdom.
G0016	APT29	APT29 is threat group that has been attributed to **Russia's** Foreign Intelligence Service (SVR). They have operated since at least 2008, often targeting government networks in Europe and NATO member countries, research institutes, and think tanks.
G0082	APT38	APT38 is a **North Korean** state-sponsored threat group that specializes in financial cyber operations; it has been attributed to the Reconnaissance General Bureau. Active since at least 2014, APT38 has targeted banks, financial institutions, casinos, cryptocurrency exchanges, SWIFT system endpoints, and ATMs in at least 38 countries worldwide.
G0073	APT19	APT19 is a **Chinese-based** threat group that has targeted a variety of industries, including defense, finance, energy, pharmaceutical, telecommunications, high tech, education, manufacturing, and legal services

Table 18: APT groups by country.

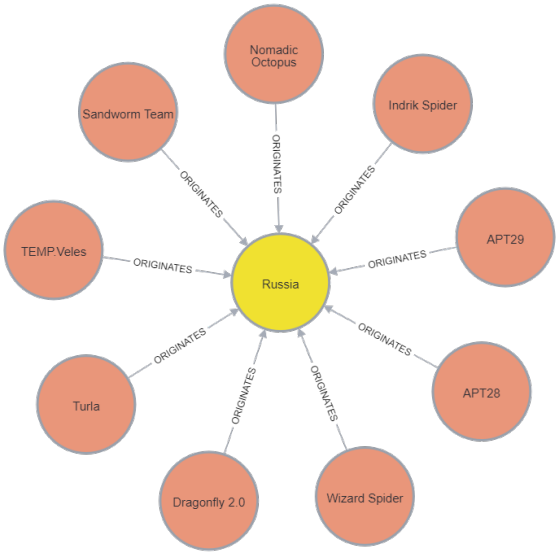
Country	**Associated Groups**
No country affiliation	56
China	33
Iran	11
Russia	9
North Korea	5
Middle East	2
Pakistan	2
South Korea	2
Lebanon	2
South America	1
India	1
Portugal	1
Spain	1
Vietnam	1
Nigeria	1

The associated knowledge graph representation for the top three country affiliations is shown in Figure 60. We note that attack groups from China, Figure 60a, are referenced three times as often as the second most mentioned region, Iran, Figure 60b.



(a) China is associated with 33 attack groups.

(b) Iran is associated with 11 attack groups.



(c) Russia is associated with nine attack groups.

Figure 60: Top three countries (yellow nodes) associated with attack groups (dark orange nodes) based on group descriptions in MITRE ATT&CK.

5.4.2 WHO ATTACKS EACH SECTOR

Second, we leveraged MITRE ATT&CK to identify adversarial groups relevant to organizations based on whom they previously targeted within similar sectors. The goal is to map attack group sectors to DHS critical infrastructure as previously discussed in Section 5.2. Again, we performed data mining using keyword phrases related to 'targets', 'targeted', or 'targeting' in the group description, as shown in Table 19, to determine a sector and country. Since the granularity was inconsistent in the attack group descriptions as previously noted, we reduced the resulting set to specifically assign the United States as a targeted country when mentioned, then grouped all other representations of locations in a single category, i.e., outside of the continental United States. This choice had no impact on our ranking framework as our focus is on attacks aimed at the United States rather than abroad.

The attribution of attack groups to sectors is shown in Table 20. Some groups, such as APT19, may target more than one sector so there is not a one-to-one relationship. We made a distinction between two sectors that are relatively similar, Government Facilities and Defense Industrial Base. Variations of 'government', 'education', and 'academic' in the description were used to categorize the Government Facilities sector. For the Defense Industrial Base sector, we sampled for the word 'defense'.

Table 19: Keywords in the description are used to determine DHS sectors and regions where the organization is located.

ID	Name	Description
G0018	admin@338	admin338 is a China-based cyber threat group. It has previously used newsworthy events as lures to deliver malware and has primarily **targeted organizations involved in financial, economic, and trade policy**, typically using publicly available RATs such as PoisonIvy, as well as some non-public backdoors.
G0099	APT-C-36	APT-C-36 is a suspected South America espionage group that has been active since at least 2018. The group mainly **targets Colombian government institutions as well as important corporations in the financial sector, petroleum industry, and professional manufacturing.**
G1000	ALLANITE	ALLANITE is a suspected Russian cyber espionage group, that has primarily **targeted the electric utility sector within the United States and United Kingdom.**
G0016	APT29	APT29 is threat group that has been attributed to Russia's Foreign Intelligence Service (SVR). They have operated since at least 2008, often **targeting government networks in Europe and NATO member countries, research institutes, and think tanks.**
G0082	APT38	APT38 is a North Korean state-sponsored threat group that specializes in financial cyber operations; it has been attributed to the Reconnaissance General Bureau. Active since at least 2014, APT38 has **targeted banks, financial institutions, casinos, cryptocurrency exchanges, SWIFT system endpoints, and ATMs in at least 38 countries worldwide.**
G0073	APT19	APT19 is a Chinese-based threat group that has **targeted a variety of industries, including defense, finance, energy, pharmaceutical, telecommunications, high tech, education, manufacturing, and legal services**

Table 20: DHS sectors ranked by the number of attack groups targeting those sectors based on mentions in MITRE ATT&CK.

Sector Name	**Groups Targeting**
Government Facilities	50
Information Technology	33
Financial Services	19
Healthcare and Public Health	17
Defense Industrial Base	14
Energy	14
Critical Manufacturing	10
Communications	9
Transportation Systems	7
Chemical	2
Water and Wastewater Systems	1
Nuclear Reactors, Materials, and Waste	1
Emergency Services	0
Dams	0
Commercial Facilities	0
Food and Agriculture	0

The methodology described in Section 5.4.1 was employed again to determine the target country using the syntax:

```
target county = targeted | targets | targeting ‘‘+’’ expression ‘‘+’’ in
            the | at ‘‘+’’ expression ‘‘+’’ country | region
```

The top 10 countries targeted by attack group are shown in Table 21. The United States is referenced as the most frequent target for adversaries. In Table 22, we see the relative number of attack groups targeting the United States originating from China, Russia, and Iran.

Table 21: Top 10 countries or regions targeted by attack groups.

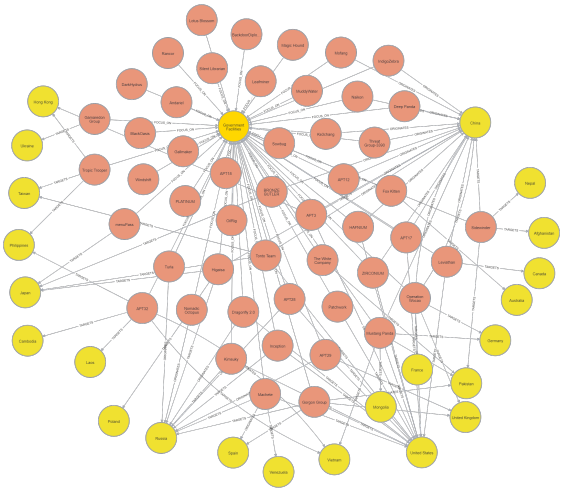
Targeted Regions	Groups Targeting
United States	24
Middle East	13
Europe	10
Russia	10
Japan	10
Asia	8
South Korea	6
Southeast Asia	6
Taiwan	5
Iran	5

The associated knowledge graph representation for the top three sector affiliations is shown in Figure 61. We see that Government Facilities, Figure 61a, are mentioned most frequently in attack group descriptions followed by Information Technology, Figure 61b, and Financial Services, Figure 61c.

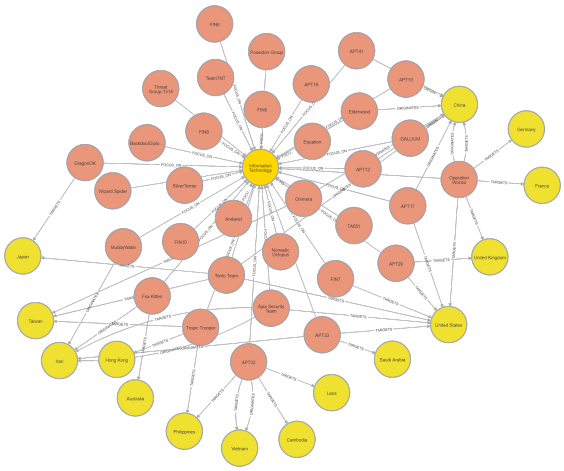
5.4.3 LINKING ATTACKER TECHNIQUES TO VULNERABILITIES

In their work which reviewed limitations and suggested enhancements based on existing threat models related to advanced persistent threat groups, Tatem et al. [151] indicated:

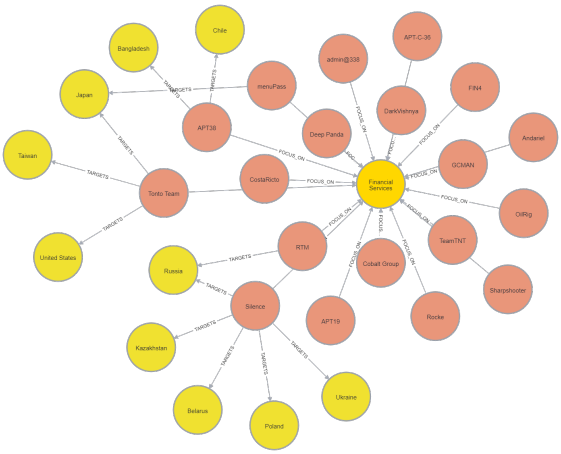
> *Based on our review, there is no single research/model that has proposed a complete solution that uses CVE information to correlate the common characteristics of attackers, exploits and vulnerabilities for attack prediction. There is also a lack of literature that attributes the actors' tactics using the mapping of CVE and CPEs.*



(a) Government Facilities sector is targeted by 50 attack groups.



(b) Information Technology sector is targeted by 33 attack groups.



(c) Financial Services sector is targeted by 19 attack groups.

Figure 61: Top three sectors (gold nodes) associated with attack groups (dark orange nodes) and their targeted country (yellow nodes) based on group descriptions in MITRE ATT&CK.

Though attack prediction is not the explicit focus of this research, we sought to complete the knowledge graph to include all relevant linkages to data points in our collected data sets.

Table 22: Operating country of attack groups targeting the United States.

Country of Attack Group	**Targeted Region**	**Groups Targeting**
China	United States	8
Unknown	United States	6
Russia	United States	3
Iran	United States	3
Middle East	United States	1
Pakistan	United States	1
North Korea	United States	1
Spain	United States	1

The 15 tactics in MITRE ATT&CK were mapped to techniques in a nonstandard format. The tactics were assigned one-to-many with each technique using the descriptive name of the tactic vice the tactic_id. In Table 23, we see that technique T1001 is used by a single tactic, while T1037 and T1053 are used by two and three tactics respectively.

Table 23: Excerpt from MITRE ATT&CK enterprise technique mapping to tactics [106].

Technique ID	**Technique Name**	**Tactic(s)**
T1001	Data Obfuscation	Command And Control
T1037	Boot or Logon Initialization Scripts	Persistence, Privilege Escalation
T1053	Scheduled Task/Job	Execution, Persistence, Privilege Escalation

We used natural language processing to extract individual tactics, then used relational database queries to assign the associated tactic_id so the resulting data would be normalized for ingestion into our knowledge graph, Table 24.

Table 24: Example of MITRE ATT&CK technique mapping after normalization [106].

Technique ID	**Technique Name**	**Tactic(s)**	**Assigned Tactic ID(s)**
T1001	Data Obfuscation	Command And Control	TA0011
T1037	Boot or Logon Initialization Scripts	Persistence, Privilege Escalation	TA0003, TA0004
T1053	Scheduled Task/Job	Execution, Persistence, Privilege Escalation	TA0002, TA0003, TA0004

With the tactic to technique mapping complete, Table 25 shows the top 20 techniques based on the overlap of attack groups utilizing that technique. We note nearly 50% or more of the 129 attack groups have used one of the top four techniques shown.

Tactics and techniques associated with attack groups provide the connection to CAPEC and ultimately to CWE (Chapter 4) as shown in Table 26. This linkage from adversary to vulnerability allows us to complete the attack pathway in our knowledge graph and will ultimately allow analysis of related CVEs and CPEs for specific organizations which will be discussed in Chapter 6.

The graph in Figure 62 provides a visual representation depicting the attack groups (i.e., 37) who use technique T1082, System and Information Discovery, and the cyber intelligence lineage we defined. Further, the completion of the knowledge graph relationships provides the traceability needed to identify attack group techniques linked to vulnerabilities known to have been exploited based on alerts in the CISA Exploit Catalog, Table 27.

Table 25: Top 20 techniques associated with attack groups.

Technique ID	**Technique Name**	**Associated Groups**
T1204.002	User Execution: Malicious File	67
T1566.001	Phishing: Spearphishing Attachment	64
T1027	Obfuscated Files or Information	62
T1105	Ingress Tool Transfer	62
T1059.001	Command and Scripting Interpreter: PowerShell	57
T1059.003	Command and Scripting Interpreter: Windows Command Shell	54
T1588.002	Obtain Capabilities: Tool	50
T1547.001	Boot or Logon Autostart Execution: Registry Run Keys / Startup Folder	47
T1071.001	Application Layer Protocol: Web Protocols	43
T1053.005	Scheduled Task/Job: Scheduled Task	38
T1059.005	Command and Scripting Interpreter: Visual Basic	38
T1070.004	Indicator Removal on Host: File Deletion	37
T1082	System Information Discovery	37
T1036.005	Masquerading: Match Legitimate Name or Location	35
T1083	File and Directory Discovery	35
T1566.002	Phishing: Spearphishing Link	34
T1005	Data from Local System	32
T1203	Exploitation for Client Execution	32
T1204.001	User Execution: Malicious Link	32
T1057	Process Discovery	31
T1003.001	OS Credential Dumping: LSASS Memory	
T1016	System Network Configuration Discovery	30
T1078	Valid Accounts	30

Table 26: Example linkage of attack group techniques from CAPEC to CWE.

Technique ID	**Technique Name**	**Associated Groups**	**CAPEC**	**CWE**
T1105	Ingress Tool Transfer	62	CAPEC-185 Malicious Software Download	CWE-494 Download of Code Without Integrity Check
T1547.001	Boot or Logon Autostart Execution: Registry Run Keys / Startup Folder	47	CAPEC-273 HTTP Response Smuggling	CWE-436 Interpretation Conflict
T1082	System Information Discovery	37	CAPEC-34 HTTP Response Splitting	CWE-74 Improper Neutralization of Special Elements in Output Used by a Downstream Component ('Injection')
T1083	File and Directory Discovery	35	CAPEC-127 Directory Indexing	CWE-276 Incorrect Default Permissions
T1005	Data from Local System	32	CAPEC-647 Collect Data from Registries	CWE-285 Improper Authorization
T1057	Process Discovery	31	CAPEC-573 Process Footprinting	CWE-200 Exposure of Sensitive Information to an Unauthorized Actor

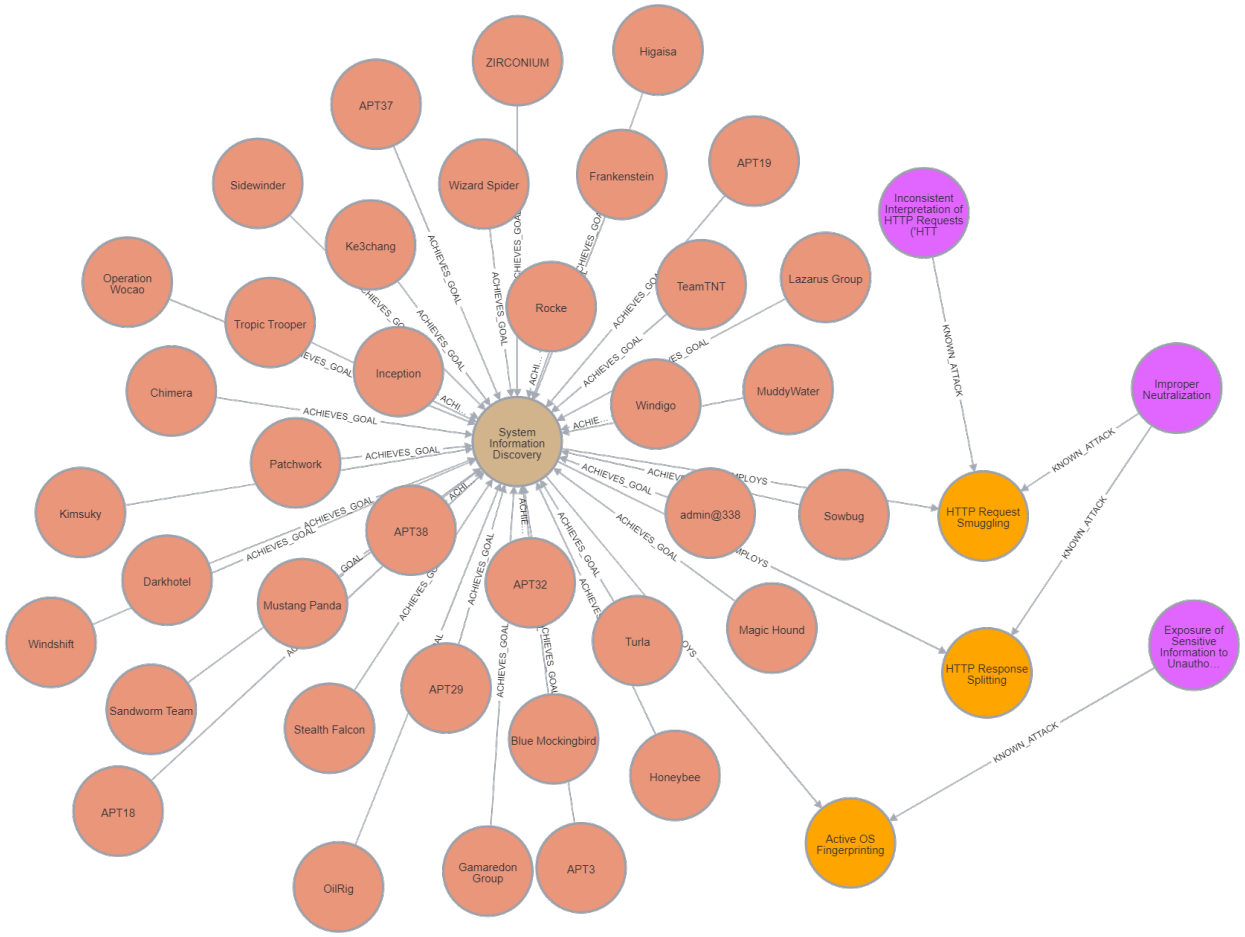


Figure 62: Attack groups (dark orange nodes) who use technique T1082 (brown node) and the associated CAPEC attack patterns (bright orange nodes) and CWE identifiers (purple nodes).

Table 27: Example using technique T1082 to demonstrate the connection to vulnerabilities and exploits.

Technique ID	**CAPEC-ID**	**CWE-ID**	**CVE-ID**	**CISA Exploit Catalog Date**
T1082	CAPEC-34	CWE-74	CVE-2019-17558	11-03-2021
T1082	CAPEC-34	CWE-74	CVE-2019-2725	01-10-2022
T1082	CAPEC-34	CWE-74	CVE-2020-17496	11-03-2021
T1082	CAPEC-34	CWE-74	CVE-2020-8644	11-03-2021
T1082	CAPEC-34	CWE-74	CVE-2021-22204	11-17-2021
T1082	CAPEC-34	CWE-74	CVE-2021-26084	11-03-2021

While, we have a priori insight of vulnerabilities already exploited, the traceability defined in our knowledge graph indicates that MITRE ATT&CK can play a major role in vulnerability prioritization. Organizations now have the requisite data needed to:

- Form a complete picture of potential threats using only publicly available cyber intelligence
- Understand the methods adversaries of interest could use to attack their networks
- Identify patterns and trends of threat actors relative to the industry and regions where the organization operates

5.5 CHAPTER SUMMARY

In this chapter, we described and analyzed the cyber intelligence datasets we used to create threat profiles for ranking cybersecurity vulnerabilities. We created an approach for characterizing threat actors based on tactics and techniques employed that fills the gap identified by Mavroeidis and Hohimer [101]. We leveraged a knowledge graph, natural language processing, and relational database query techniques to standardize and supplement the underlying cyber intelligence to show that CVE information in the NVD can be correlated with known attackers, exploits and vulnerabilities for attack prediction. Finally, we presented a complete graph-based solution that addresses the cybersecurity challenge noted

by researchers Tatem et al. [151] to link a threat actor's tactics using the mapping of CVE and CPEs. Knowing the MITRE ATT&CK tactics and techniques for a given vulnerability, the attack groups who leverage those techniques, and the critical infrastructure sectors they target allows for the development of a more proactive prioritization strategy that focuses on the vulnerabilities that matter the most.

Chapter 6

RELEVANCE RANKING FRAMEWORK

"Mission-critical information assets – an organisation's "crown jewels" – are information assets of greatest value and would cause major business impact if compromised. These assets attract the attention of highly capable adversarial threats, all of whom are intent on exploiting this valuable information." [80]

In the past few years, CISA issued Emergency Directives (Chapter 1) to focus their efforts towards remediating vulnerabilities and active exploits determined to carry significant risk [40]. CISA observed that "risk scores based on the Common Vulnerability Scoring System (CVSS) does not always accurately depict the danger or actual hazard that a CVE presents." [42] Attackers do not rely on critical vulnerabilities to achieve their goals, but may instead exploit vulnerabilities of choice. In November 2021, CISA issued BOD 22-01, which instructed federal agencies to steer away from focusing solely on CVSS scores and instead to "target vulnerabilities for remediation that have known exploits and are being actively exploited by malicious cyber actors." [42] This BOD clearly establishes the need and provides direction for readily accessible ranking policies like the ones we developed in this study (Section 6.2).

In this chapter, we describe a framework that utilizes the aggregated data sets and knowledge graph described in Chapter 5. Our goal was to define an approach to cybersecurity vulnerability mitigation that improves upon rankings that employ strategies based on the global CVSS metrics associated with known software vulnerabilities published in the NVD. The outcome of this study is a set of relevance-based ranking models that can be employed before an adversary (i.e., cyber threat) takes advantage of a particular vulnerability. This chapter addresses work completed to address RQ2:

RQ2: What are the characteristics needed to define vulnerability ranking policies that improve the return on investment (ROI) of applied mitigations, compared to traditional CVSS Base Score policies, relevant to the organization's specific mitigation goals and priorities?

The architecture for our relevance model requires the following components:

- Profiles that describe the organization under evaluation in terms of the DHS sector and country in which they operate (Section 6.1.1, 6.1.3)

- Collection and normalization of a complete software inventory for each organization (Section 6.1.2)
- Threat-centric ranking policy definitions based on attack groups of interest and their skill level (Section 6.2)
- Scoring method for each ranking policy (Section 6.3)

6.1 CREATING ORGANIZATIONAL PROFILES

A vulnerability ranking policy needs to first consider the attack surface (i.e., installed software) for the organization under evaluation. Table 20 (in Chapter 5) shows Government Facilities are the most targeted DHS critical infrastructure sector based on MITRE ATT&CK attack group descriptions. Therefore, for this study, we sought to identify and define a representative set of organizations in the Government Facilities to serve as the test set for developing and evaluating our vulnerability management approach. We will include the Education subsector for completeness.

6.1.1 SOFTWARE USED IN THE EDUCATION SUBSECTOR

The website CollegeSimply [32] provides a list *Virginia Colleges Ranked by Largest Enrollment*. The website sources public domain college data from the U.S. Department of Education National Center for Education Statistics. Using the CollegeSimply list, we identified universities of various sizes, public and private, in an attempt to have software diversity among the organizations chosen as real world examples for the Education subsector. For reference, the enrollment headcount for each university is shown in Table 28.

Table 28: Enrollment for Virginia universities [32].

	W&M	**ODU**	**VT**	**RU**	**UVA**	**WLU**
Enrollment	8,939	24,286	37,024	10,483	25,628	2,183

The availability of a supported software list was the criteria used to narrow down the list of Virginia Colleges to six candidates of various sizes for our study:

- William and Mary (W&M)
- Old Dominion University (ODU)
- Regent University (RU)
- Virginia Tech (VT)
- University of Virginia (UVA)
- Washington and Lee University (WLU)

In the absence of network discovery scans that perform software enumeration for real-world organizations, we searched each university's web site to locate a published list of supported academic software as shown in Table 29. The software lists are normally maintained by the school's Information Technology department and publicly available. We used web scraping techniques to retrieve the associated software list from each website.

Table 29: Websites used to identify academic software lists.

University	**Source for Software List**
RU	`https://www.regent.edu/information-technology/#supported-software`
W&M	`https://software.wm.edu/all-software/`
ODU	`https://www.odu.edu/ts/software-services#tab110=1`
UVA	`https://virginia.service-now.com/its?id=software_gateway`
WLU	`https://my.wlu.edu/its/services/academic-technologies/our-services`
VT	`https://itpals.vt.edu/softwarelicensingcenter/studentsoftware/studentswproductlist.html`

6.1.2 ASSIGNING CPE IDS

Section 4.1.1 described the vendor product data set (i.e., CPE) and how it is used to identify CVE-IDs. Table 30 shows an excerpt of the product names as they appeared on each university's website. The full academic software listing is provided in Appendix A, Table 48. Since these lists consist of product names only, we need to determine a CPE-ID in order to dynamically assign vulnerabilities using graph queries. The data was found to contain similarities in software usage as might be expected for organizations in the same sector. For example, five of six universities listed *Zoom* and every university used one or more products sold by *Adobe* or *Microsoft*. Disparities in some product names required standardization before a CPE-ID could be assigned. A Google search, using the product name as listed, was used to resolve naming inconsistencies which included:

- Products without a vendor name (e.g., Matlab should be Mathworks Matlab)
- Abbreviated software name (e.g., MSC Patran/Nastran should be MSC Software)
- Product names changed by the vendor (e.g., WebXtender should be Ellucian Banner Web Tailor)
- Vendor without a product name (e.g., Zoom should be Zoom Meetings)

A Google search was also used to identify the vendor's preferred name for their software which we converted to the part, vendor, product standard notation using CPE nomenclature (Section 2.2.1, Figure 16). Collected data specific to academic organizations was used to analyze vulnerabilities.

Table 30: Excerpt from the software lists of Virginia universities.

W&M	**ODU**	**VT**	**RU**	**UVA**	**WLU**
Adobe Acrobat Reader DC	7-Zip	Adobe Creative Cloud	Adobe Acrobat Reader Dc	Adobe Fonts	After Effects
Advance	AGI/ STK11	Autodesk	Adobe Acrobat Pro Dc	Adobe Reader	Illustrator
Alertus Desktop Notification	Adobe Acrobat Professional	Duo D-100 Token	Adobe Creative Cloud	Alertus	Photoshop
AutoDesk	Adobe After Effects CC	ESRI ArcGIS Desktop Education Edition	Adobe Spark	ANSYS	Premiere
Banner Admin Production	Adobe Animate	EquatIO	Application Xtender	ArcGIS Suite	Audacity
Box	Adobe Creative Cloud	GRAHL PDF Annotator	Autodesk Maya	Dedoose	Autodesk Inventor
Google Chrome	ArcGIS	Rhino	Microsoft Office	MetaAccess	FlipGrid
LinkedIn Learning	Arena	SAS	Microsoft Project	Microsoft Defender for Endpoint	iMovie
Minitab for Windows	ERDAS Imagine	Windows Virtualization Add-on License	Mozilla Firefox	Microsoft Software & Tools for Learning	Wordpress

The NVD provides a search engine[1] that performs a keyword search on all components of the CPE-ID. We entered the part, vendor, product into the search engine to identify a CPE-ID for each instance of academic software. For each software product, we captured the most recent version number based on our assumption that IT departments perform due diligence, practice disciplined cyber hygiene, and upgrade to the latest software versions as they become available. The use of standard naming and the CPE-ID search resulted in a consolidated list of 83 software products across all universities. An excerpt is shown in Table 31.

Table 31: Example of software standardization for assignment of CPE-IDs.

Software	**Part**	**Vendor**	**Product**
Google Calendar	a	jenkins	google_calendar
CPE ID	cpe:2.3:a:jenkins:google_calendar:0.2:*:*:*:*:jenkins:*:*		
MATLAB	a	mathworks	matlab
CPE ID	cpe:2.3:a:mathworks:matlab:7.5:*:*:*:*:*:*:*		
SPSS Statistics (Teaching and Research)	a	ibm	spss_analytic_server
CPE ID	cpe:2.3:a:ibm:spss_analytic_server:3.1.2:*:*:*:*:*:*:*		
WebXtender	a	ellucian	banner_web_tailor
CPE ID	cpe:2.3:a:ellucian:banner_web_tailor:8.9:*:*:*:*:*:*:*		
Adobe Reader	a	adobe	acrobat
CPE ID	cpe:2.3:a:adobe:acrobat:_reader9.5.3:*:*:*:*:*:*:*7		

Some academic software could not be assigned a CPE-ID based on results of the CPE search. Items in this category were primarily specific research tools with a unique purpose (e.g., video editing, statistical packages) or a vendor not represented in the NVD. Software without a CPE-ID was excluded from the consolidated list, not assigned to a university, and was not subjected to the relevance ranking policies in Section 6.2. Table 32 shows

[1] `https://nvd.nist.gov/products/cpe/search`

the number of published software for each university along with the number for which we could identify a CPE-ID. In this study, we added a size designation to represent a small, medium, large, and extra large organization where size relates to installed software vice enrollment or another metric. We note that Old Dominion University provided the most fidelity in their reporting with 69 identified software products. In the cases of Virginia Tech and Washington and Lee, we suspect their software lists are more robust than indicated on their respective websites and is likely under-reported. We contend any vulnerability ranking policy will require some degree of accurate self reporting to properly determine the software inventory.

Table 32: Academic software associated with vendor product CPE-ID.

	W&M	**ODU**	**VT**	**RU**	**UVA**	**WLU**
CPEs Assigned	24	47	12	23	30	13
Software Listed	33	69	22	31	49	23
Size Designation	Medium	Extra Large	Small	Medium	Large	Small

6.1.3 SOFTWARE USED BY GOVERNMENT FACILITIES

Government facilities do not routinely publish software they use on a public website. There is, however, a policy to ensure installed software is vetted and approved for use. The National Information Assurance Acquisition Policy, NSTISSP No. 11 [3], requires government agencies to purchase only those commercial security products that have met specified third-party assurance requirements and have been tested by an accredited national laboratory [3].

In accordance with NSTISSP, the *Common Criteria* is an internationally recognized set of guidelines (ISO 15408) [81] that define a common framework for evaluating security features and capabilities of Information Technology security products against functional and assurance requirements. Once certification is completed, it provides assurance to government buyers that the process of specification, implementation, and evaluation for any certified

computer security solution was conducted in a thorough and standard manner. The list of certified products[2] can be downloaded in a CSV file format. The metadata shown in Table 33 was extracted from the Common Criteria download file.

Table 33: Common Criteria feature extraction.

Field	**Example**
Product	Oracle Database 12c Release 1 Enterprise Edition, Version 12.1.0.2
Vendor	Oracle Corporation
Scheme	US

The complete list contained 2000 products across 15 categories, Table 34. A certified product can be associated with multiple categories [81]. We reduced Common Criteria to just the set of products certified for use in the U.S. scheme (i.e., 181 products) based on our finding that the United States is the most targeted region by adversaries (Section 5.4.2). As described in Section 6.1.2, we searched for CPE-IDs across all categories based on the vendor and product name which further reduced the list to 57 products eligible for inclusion in this study.

[2] `https://www.commoncriteriaportal.org/products/`

Table 34: Certified products by category [81].

Category	**Products**
Access Control Devices and Systems	22
Biometric Systems and Devices	0
Boundary Protection Devices and Systems	43
Data Protection	61
Databases	14
Detection Devices and Systems	9
ICs, Smart Cards and Smart Card-Related Devices and Systems	588
Key Management Systems	10
Mobility	30
Multi-Function Devices	232
Network and Network-Related Devices and Systems	230
Operating Systems	53
Other Devices and Systems	270
Products for Digital Signatures	60
Trusted Computing	37

We attempted to mirror the organizations in the Education subsector by creating four government organizations to serve as proxies for real-world institutions. The government organizations were designated as *GOV-S, GOV-M, GOV-L, and GOV-XL* to represent a small, medium, large, and extra large organization where size relates to installed software vice number of employees or another metric. The software list shown in Table 35, consisting of applications and operating systems, was generated by randomly selecting software from Common Criteria with assigned CPE-IDs in groups of 14, 24, 30, and 47 to approximately match the cardinality of the university software lists, Table 32.

Table 35: Government facility software associated with vendor product CPE-ID.

	GOV-S	GOV-M	GOV-L	GOV-XL
Software Assigned	14	24	30	47
Common Criteria	57	57	57	57

Table 36 shows an excerpt from the randomly generated software list for Government Facilities using Common Criteria as the source. The full software listing is provided in Appendix B, Table 49.

6.2 RANKING POLICY DEFINITIONS

Deciding which vulnerabilities to remediate is a daunting task. In a perfect world, all vulnerabilities would be remediated as they were discovered, but unfortunately that does not happen in reality. With thousands of new vulnerabilities every year multiplied across disparate assets, reality necessitates prioritization. It comes down to choosing a subset of vulnerabilities to focus on first. But how can we measure the quality of prioritization? Because an exploit observed in the wild is the most relevant proxy for the probability that an exposed vulnerability can be used to compromise an organization's network, this study focused on building predictive ranking policies to identify candidate vulnerabilities that fit the pattern of known attack groups. A key metric is the overlap between vulnerabilities in the software used by an organization and the ones being actively targeted by threat actors.

Two approaches exist for organizing knowledge about adversary behavior, CAPEC (Section 2.2.2) and MITRE ATT&CK (Section 2.2.3), each focused on a specific set of use cases. CAPEC describes the common attributes and techniques employed by adversaries to exploit known weaknesses. ATT&CK details the specific tactics, techniques, and procedures that advanced persistent threats (APT) use to execute their objectives. Each attack pattern captures knowledge about how specific parts of an attack are designed and executed and provide guidance on ways to mitigate the attack's effectiveness.

CAPEC differs from the MITRE ATT&CK framework in the way it focuses on application security and enumerates exploits against vulnerable systems, while the MITRE ATT&CK framework focuses on network defense and provides a contextual understanding

Table 36: Excerpt from the generated software list for government facilities.

GOV-S	**GOV-M**	**GOV-L**	**GOV-XL**
aruba 2930f	application delivery controller	altalink c8070 firmware	adaptive security appliance
cloud access manager	aruba 2930f	anyconnect secure mobility client	advanced threat defense
core	carbon black app control	catalyst ie3200 rugged switch	altalink c8070 firmware
extremexos	cloud access manager	clearpass	anyconnect secure mobility client
gigavue	core	cloud access manager	avocent umg-4000 firmware
globalprotect	cx 6200f	core	carbon black app control
knox	cx 8320	data protector	catalyst ie3200 rugged switch
m-200	cyber backup	desktop password reset	clearpass
securitycenter	display solutions	dm-nvx-dir-160	cloud services router 1000v
unified endpoint management	enterprise linux eus	enterprise linux eus	clustered data ontap

of malicious behavior. Both frameworks have utility in application threat modeling, comparing network defense capabilities, identifying new threats, and provide the foundation for our relevance ranking policies.

We defined the criteria for developing our ranking policies using the predominant attacker characteristics and targets discussed in Chapter 5. Table 18 identified China, Iran, and Russia as the top three operating regions for attack groups. Table 20 identified Government Facilities as the most targeted DHS critical infrastructure sector. Table 21 identified the United States as the most frequent target for adversaries. We use this information to create an ideal ranking policy based on specific threat groups and known exploits (Section 4.1.5) and a more generalized approach which considers the skill level of the attacker and impact to the organization as represented in CAPEC (Section 4.1.4). Each policy (Section 6.2) leveraged data points in the knowledge graph to provide a scoring methodology that considers:

- Which threat actors are using the same technique to launch an attack (Section 5.4)?
- Which threat actors actors target our industry (Section 5.4.2)?
- Which vulnerabilities does this type of attack exploit (Section 5.4.3)?
- What is the imminent probability of exploit (Section 4.1.5)?
- Which vulnerabilities are present in our organization (i.e., installed software) (Section 6.1)?

Next, we created four ranking policies to provide context for the relevance ranking schemes we examined and evaluated using Normalized Discounted Cumalitive Gain (nDCG) [87] as discussed in Chapter 7. The primary consideration when using nDCG is that we do not know the ideal ranking of vulnerabilities that could be discerned based on preferences of the organization. Ideally, an organization only wants to patch the vulnerabilities that might actually be exploited to cause a security incident. Patching anything else is unnecessary work. Therefore, in this study we developed ranking policies based on information in the content of CVE-IDs rather than on other opinions. Specifically, we assert the ideal ranking is one which considers the preferences of the organization with regard to specific threats (Policy Two, Policy Three) and includes knowledge on whether the CVE-ID has already been exploited (Policy Four). The presence of a personalized ideal ranking allowed us to individually measure the performance and quality of each ranking methodology we proposed.

- **Policy One: CVSS Base Score Ranking**: The organization remediates based on the assigned CVSS Base Score ranking from most severe ("critical") to least severe ("low").
- **Policy Two: APT Threat Ranking**: The organization is most concerned about predicting vulnerabilities with a high likelihood of imminent exploit that can be exploited by a technique employed by an attack group referenced in Table 18 that targets the industry in the country where the organization operates.
- **Policy Three: Generalized Threat Ranking**: The organization is concerned about predicting vulnerabilities with a high likelihood of exploit by a low- or highly-skilled adversary that have high impact on the organization.
- **Policy Four: Ideal Ranking**: The ideal ranking employs the same criteria as the APT and Generalized threat rankings, Policy Two and Three, but has the foreknowledge that a vulnerability has already been exploited using information from the ExploitDB and CISA KEV databases.

For all ranking policies, the set of applicable CVE-IDs are delivered in ascending order by relevance score, then subsequently ordered by CVE-ID to avoid ties.

6.3 IMPLEMENTATION

In the first stage, each ranking model starts from a potentially huge corpus of vulnerabilities which are published daily and generates a smaller subset based on the installed software at a particular institution (Section 6.1.1, 6.1.2). This approach is consistent with industry practices as we would not expect an organization to expend energy to evaluate CVE-IDs related to Redhat Linux, for example, when all of their servers use a Windows operating system. Since our knowledge graph includes all product versions as listed in the CPE Dictionary, we use graph queries to determine the latest software version during query execution.

6.3.1 FEATURE SELECTION

For each CVE-ID, we examined 15 features using the cyber intelligence data sources identified in Section 2 and requirements dictated by the policy definitions (Section 6.2). The features enumerated here provide the insight needed to inform each policy and create a set of relevance scores for ranking CVE-IDs as they are published.

1. CVE-ID
2. CVSS V3.1 Base Score
3. CVSS Base Score Metrics
4. Publication date
5. Modification date
6. CAPEC-ID
7. CAPEC skill level
8. ATT&CK technique name
9. ATT&CK group id
10. ATT&CK group country of operation
11. Risk Appetite
12. EPSS Probability of Exploit
13. CISA Known Exploit Catalog
14. Exploit-DB
15. Organization identifier
16. Critical infrastructure sector
17. Organization's country of residence

Based on the policy definitions, the CVSS V3.1 Base Score is the only feature needed to implement Policy One. The features needed to implement Policy Two and its ideal version in Policy Four are listed in Table 37.

Table 37: Policy Two scoring features using MITRE ATT&CK data feed to characterize the threat to the organization.

Feature	**Specific Threat Relevance Rank Value**	**Ideal Rank Value(s)**
CVSS Base Metric (Attack Vector)	Network	Network
DHS Sector	Government Facilities Education	Government Facilities Education
Org Country	United States	United States
Attack Group Country	China, Russia, Iran	China, Russia, Iran
Risk Appetite	[0, 100]	[0, 100]
EPSS Probability	0.876	
CISA KEV or Exploit-DB Entry Exists		True
Software Affected	True	True
Scoring Range	[1-6]	[1-6]

For Policies Two, Three, and Four, we established a binary weighting [0,1] for each feature to determine its existence as applicable to a specific CVE-ID. The sum of the categorical values is presented as the relevance score we used to rank the associated CVE-IDs using the logic shown in Algorithm 1. The minimum assigned relevance score is set to 1 using this algorithm to avoid a long tail of non-relevant CVE-IDs and ensure only relevant CVE-IDs associated with the organization's installed software are candidates for ranking.

6.3.2 RANK CANDIDATE VULNERABILITIES

Prior to ranking Relevance Policy Three, we converted the ordinal entries for CAPEC likelihood of attack, CAPEC severity, and CAPEC skill level to numeric scores, as follows:

- Likelihood of attack is represented as low, medium, and high in the CAPEC dictionary. We converted these entries to 1, 2, and 3 respectively.

- Severity is represented as low, medium, high, and very high in the CAPEC dictionary. We converted these entries to 1, 2, 3, and 4 respectively.
- Skill level is represented as low, medium, and high in the CAPEC dictionary. We converted these entries to 1, 2, and 3 respectively noting in Policy Three the organization wants increased attention paid to vulnerabilities that can be exploited by an unskilled adversary (i.e., low skill level) and a highly-skilled one (i.e., attack group).

The features needed to implement Policy Three and its ideal version in Policy Four are listed in Table 38. The sum of the categorical values is presented as the relevance scored we used to rank the associated CVE-IDs using the logic shown in Algorithm 2. Again, the minimum assigned relevance score is set to 1 using this algorithm to avoid a long tail of irrelevant CVE-IDs and ensure only relevant CVE-IDs associated with the organization's installed software are candidates for ranking.

Table 38: Policy Three scoring features using CAPEC data feed to characterize the threat to the organization.

Feature	**General Threat Relevance Rank Value**	**Ideal Rank Value(s)**
CVSS Base Metric (Attack Vector)	Network	Network
DHS Sector	Government Facilities	Government Facilities
	Education	Education
Likelihood of Exploit	High	High
Impact Severity	High	High
Attacker Skill Level	Low	Low
	High	High
Risk Appetite	[0, 100]	[0, 100]
EPSS Probability	0.876	
CISA or POC Exists		True
Software Affected	True	True
Scoring Range	[1-7]	[1-7]

Algorithm 1 APT Threat Relevance Rank Scoring Model.

Let $org \leftarrow organization\_id$ ▷ Section 6.1
Let $gc \leftarrow attackGroupCountry$ ▷ China, Russia, or Iran
Let $ra \leftarrow riskAppetite$ ▷ correlates to EPSS percentile
Let $wsd \leftarrow weekStartDate$
$calcRelevanceScore(org, gc, ra, wsd)$
procedure CALCRELEVANCESCORE(org, gc, ra, wsd)
    $(orgCountry, DHSsector, installedSW) \leftarrow OrgProfile(org)$ ▷ Section 6.1
    $OrgCve\_ids \leftarrow getWeeklyVulns(org, wsd, installedSW)$
    for all $o \in OrgCve\_ids$ **do**
        $(idealScore, relScore) \leftarrow 0$
        $(idealScore, relScore) \leftarrow scenarioATT\&CK$
    end for
end procedure
function SCENARIOATT&CK
    if $cpeID(installedSW) \in cpeID(cveSoftware)$ **then** ▷ Section 6.1.2
        Increment $relScore$ **and** $idealScore$
        $attackVector \leftarrow CVSSBaseScoreMetrics(o.Cve\_id)$
        if $(attackVector == `NETWORK')$ **then** ▷ Remotely exploitable
            Increment $relScore$ **and** $idealScore$
        end if
        $cisa \leftarrow cisa\_exists(o.Cve\_id)$
        $exploitdb \leftarrow exploitdb\_exists(o.Cve\_id)$
        if ($cisa$ **or** $exploitdb$) **then** ▷ Known exploit or POC exists
            $idealScore := idealScore + 1$
        end if
        $percentile \leftarrow EPSSPercentile(o.Cve\_id)$ ▷ Probability of exploit
        if $(percentile >= ra)$ **and** (**not** $cisa$) **then** ▷ Not already exploited
            $relScore := relScore + 1$
        end if

```
30:         threat ← knownThreat(o.Cve_id, gc)          ▷ Group techniques target CVE_ID
31:         if threat then
32:             Increment relScore and idealScore
33:             if country(o.ve_id) = gc then                          ▷ Group country
34:                 Increment relScore and idealScore
35:             end if
36:             sectorThreat := sectorThreatExists(o.Cve_id, country, DHSSecctor
37:             if sectorThreat then                                ▷ Organization's Sector
38:                 Increment relScore and idealScore
39:             end if
40:             if  sectorCountry = country then                   ▷ Organization's Country
41:                 Increment relScore and idealScore
42:             end if
43:         end if
44:         return relScore, idealScore
45:     else
46:         return                          ▷ No match on CVE-ID, skip scoring
47:     end if
48: end function
```

Algorithm 2 General Threat Relevance Rank Scoring Model

```
1: Let org ← organization_id                                   ▷ Section 6.1
2: Let gc ← attackGroupCountry                                 ▷ China, Russia, or Iran
3: Let ra ← riskAppetite                                       ▷ correlates to EPSS percentile
4: Let wsd ← weekStartDate
5: calcRelevanceScore(org, gc, ra, wsd)
6: procedure calcRelevanceScore(org, gc, ra, wsd)
7:     (orgCountry, DHSsector, installedSW) ← OrgProfile(org)  ▷ Section 6.1
8:     OrgCve_ids ← getWeeklyVulns(org, wsd, installedSW)
9:     for all o ∈ OrgCve_ids do
10:        (idealScore, relScore) ← 0
11:        (idealScore, relScore) ← scenarioCAPEC
12:    end for
13: end procedure
14: function scenarioCAPEC
15:    if cpeID(installedSW) ∈ cpeID(cveSoftware) then          ▷ Section 6.1.2
16:        Increment relScore and idealScore
17:        attackVector ← CVSSBaseScoreMetrics(o.Cve_id)
18:        if (attackVector == 'NETWORK') then                 ▷ Remotely exploitable
19:            Increment relScore and idealScore
20:        end if
21:        cisa ← cisa_exists(o.Cve_id)
22:        exploitdb ← exploitdb_exists(o.Cve_id)
23:        if (cisa or exploitdb) then                         ▷ Known exploit or POC exists
24:            idealScore := idealScore + 1
25:        end if
```

```
        percentile ← EPSSPercentile(o.Cve_id)                    ▷ Probability of exploit
        if (percentile >= ra) and ( not cisa) then               ▷ Not already exploited
            relScore := relScore + 1
        end if
        threat ← knownThreat(o.Cve_id, gc)          ▷ Group techniques target CVE_ID
        if threat then
            Increment relScore and idealScore
            if country(o.ve_id) = gc then                        ▷ Group country
                Increment relScore and idealScore
            end if
            sectorThreat := sectorThreatExists(o.Cve_id, country, DHSSecctor
            if sectorThreat then                                 ▷ Organization's Sector
                Increment relScore and idealScore
            end if
            if  sectorCountry = country then                     ▷ Organization's Country
                Increment relScore and idealScore
            end if
        end if
        return relScore, idealScore
    else
        return                                  ▷ No match on CVE-ID, skip scoring
    end if
end function
```

6.3.3 ECONOMICS OF PATCH PRIORITIZATION

When determining what to patch, an IT manager must consider both the setup and business disruption costs, weigh them against the potential exploitation cost, and decide when and how often to patch an enterprise system or application. This is the trade-off, or ROI, we want to analyze using a rigorous quantitative model. For the purpose of this study we note that vulnerability response processes can differ among characteristics like response time, involved roles, impact on or disruption of production operations, and ultimately the total cost of remediation. As noted in Section 3.2.1, Fhurwith et al. [61] defined the total costs of a vulnerability response process as the sum of its direct costs (level of effort employed by human resources) and indirect costs (productivity losses, interruption of production processes after patching). Fhurwith et al. further assigned cost factors to vulnerabilities to measure execution of the response in non-monetary units based on the severity score where Low=0.25, Medium=1, High=1.5, and Critical=3 units. Similarly, we evaluated the number of vulnerabilities remediated to measure the savings realized using the same relative values.

6.3.4 TESTING AND EVALUATING

For testing and evaluating the ranking policies we used the organizational profiles and associated software lists defined in Section 6.1 to identify applicable CVE-IDs, then scored and ranked them using the algorithms defined in Section 6.2. We also established a minimum threshold for the number of vulnerabilities to be evaluated in a given week. We will evaluate the ranking of the recommendations against the ideal ranking, Policy Four, using nDCG. This will be discussed further in Chapter 7.

6.4 CHAPTER SUMMARY

In this chapter, we presented the baseline of our work and showed the main steps starting from creating an organizational profile to identify the software inventory to prioritizing vulnerabilities based on the likelihood of exploit by known attack groups. We reviewed the applicable research question and presented a detailed overview of how to define a ranking policy. The first approach is based on determining the features in our knowledge graph which allow us to determine the likelihood of exploit by a known adversary. The second approach attempts to predict an adversary's next target by formulating attack scenarios. The main steps include mapping publicly available cyber intelligence to an ontology in the knowledge graph, filtering candidate CVE-IDs, and ranking those candidates using a

personalized approach. We described each step and presented some solutions to challenges that may occur when working with non-standard data sets. Finally, we determined the methods for testing, evaluating, and quantitatively determining the return on investment for mitigating in accordance with the defined ranking policies using nDCG and patch costs. This will be discussed within the context of operational examples in Chapter 7.

Chapter 7

FRAMEWORK EVALUATION

The contributions of the previous chapters provide the foundation for this chapter in which we evaluate our framework for ranking vulnerabilities utilizing a threat-centric approach. First, we evaluate each ranking policy (Section 6.2) using the organizational profiles for Government Facilities (Section 6.1.1) and the Education subsectors (Section 6.1.3) along with techniques employed by threat actors associated with China, Russia, and Iran (Sections 5.4.2, 5.4.3) as presented in MITRE ATT&CK. Next, we determine the cost of each ranking policy by assigning patch costs to each CVE-ID using the measures defined in Fruhwith et al. [61]. Finally, we show the overall relevance scores and the results when applying each policy against a particular type of generalized and specific threat.

7.1 CANDIDATE GENERATION

In our study, we used 55,939 CVE-IDs published between 2019 and 2021 as the corpus from which to identify a much smaller subset of candidate vulnerabilities for ranking. We used the CVE modification date to examine vulnerabilities as they were published weekly using Microsoft's Patch Tuesday as the starting day for weekly collection. CVE-IDs in each week were selected based on the latter of the published date versus the modification date shown in the NVD. Some CVE-IDs may reappear in queries for subsequent weeks if they were re-analyzed by NVD analysts or new information (e.g., vendor advisory, exploit) was added to the listed CVE details, thus resulting in an updated modification date. A total of 13,862 CVE-IDs were found to be applicable across all of the Government Facilities (Table 39) and Education subsector (Table 41) software lists after matching vulnerabilities based on the CPE IDs (Section 6.3.1). The candidate list of vulnerabilities included 3,079 unique CVE-IDs.

For the Government Facilities shown in Table 39, we observed low, annual vulnerability counts for three of the four proxy organizations at less than 2% of all CVE-IDs analyzed. Even the largest government organization, GOV-XL, which was designed to mirror the breadth of software (i.e., 47 products) of its counterpart Old Dominion University (ODU) in the Education subsector, experienced less than 4% of all CVE-IDs analyzed. The low number of vulnerabilities in the sector may be attributed to the selection process for software

products assigned to Government Facilities in this study which were selected exclusively from the certified product list approved by the *Common Criteria* [3]. This outcome may provide an indication that the rigor imposed upon these products in terms of security requirements and on-going evaluation may potentially reduce their exposure to published vulnerabilities.

Table 39: Total vulnerabilities by year for government facilities sector.

Year	**GOV-S**	**GOV-M**	**GOV-L**	**GOV-XL**
2019	8	41	51	102
2020	11	34	55	144
2021	16	84	140	285
Total Vulns	35	159	246	531
Pct of All Vulns	0.25%	1.15%	1.77%	3.85%

Further, in Table 40, we observed the majority of organizations routinely have only two to four CVE-IDs to contend with week after week. We also observed the trickle rate at which vulnerabilities appeared over the course of the year for smaller organizations (e.g., GOV-S) was sufficiently low enough to perhaps not warrant a specialized ranking policy. For example, GOV-S encountered an average of two vulnerabilities per week between 2019 and 2021, but only during a small fraction (approximately 10-20%) of the calendar year, leaving ample opportunity for IT administrators to pursue other duties during the off weeks.

Table 40: Weekly vulnerability traffic by year for government facilities sector.

Year	Org	Avg Vuln Per Week	Min Vuln Per Week	Max Vuln Per Week	Weeks Per Year
2019	GOV-XL	4	1	20	32
2019	GOV-L	3	1	9	24
2019	GOV-M	2	1	4	23
2019	GOV-S	2	1	2	5
2020	GOV-XL	4	1	13	40
2020	GOV-L	3	1	10	23
2020	GOV-M	3	1	10	16
2020	GOV-S	2	1	3	6
2021	GOV-XL	7	1	25	43
2021	GOV-L	4	1	14	40
2021	GOV-M	3	1	11	29
2021	GOV-S	2	1	3	10

For the Education subsector shown in Table 41, we again observed low vulnerability counts of less than 2% for small organizations such as Virginia Tech (VT) and Washington and Lee University (WLU) who did not self-report their academic software inventory with high fidelity on their respective university websites. Conversely, we note that universities who reported more software in use such as ODU, Regent University (REGENT), and William and Mary (WM) will need to evaluate hundreds of vulnerabilities as candidates for remediation during the course of any given year.

Table 41: Total vulnerabilities by year for education subsector.

Year	ODU	REGENT	UVA	VT	WLU	WM
2019	1457	1396	188	14	3	1370
2020	751	565	279	6	57	556
2021	2026	1721	639	15	144	1704
Total Vulns	4234	3682	1106	35	204	3630
Pct of All Vulns	30.54%	26.56%	7.98%	0.25%	1.45%	26.19%

Finally, in Table 42, we note that WM, ODU, and REGENT experienced a steady stream of vulnerabilities across all three years of this study. They also experienced an increase in the number of weeks per year during which a continuous remediation policy would be advantageous. For ODU in particular, we note an increase from 42 weeks per year in 2019 to 50 weeks per year in 2021. There were several weeks in which a marked increase over the average number of vulnerabilities was observed across organizations. An explanation was not available in the NVD for large spikes we observed such as:

- 400+ applicable CVE-IDs identified during the week of 20-Aug-2019
- 300+ applicable CVE-IDs identified during the week of 20-Jul-2021
- 200+ applicable CVE-IDs identified during the week of 7-Sep-2021

Table 42: Weekly vulnerability traffic by year for education subsector.

Year	Org	Avg Vuln Per Week	Min Vuln Per Week	Max Vuln Per Week	Weeks Per Year
2019	WM	43	1	441	32
2019	ODU	35	1	444	42
2019	REGENT	35	1	442	40
2019	UVA	11	1	44	18
2019	VT	2	1	4	7
2019	WLU	1	1	1	3
2020	ODU	18	1	59	43
2020	REGENT	15	1	57	40
2020	WM	15	1	58	39
2020	UVA	9	1	34	33
2020	WLU	5	1	20	12
2020	VT	1	1	1	6
2021	ODU	41	1	315	50
2021	REGENT	36	1	264	48
2021	WM	36	1	258	48
2021	UVA	15	1	120	43
2021	WLU	7	1	23	21
2021	VT	3	1	4	7

Figures 63 and 64 show the accumulated vulnerabilities by month and year for each organization in our study. In both Figure 63 and Figure 64, we note the sporadic and unpredictable manner in which newly published and modified CVE-IDs applicable to an organization present themselves for analysis and remediation.

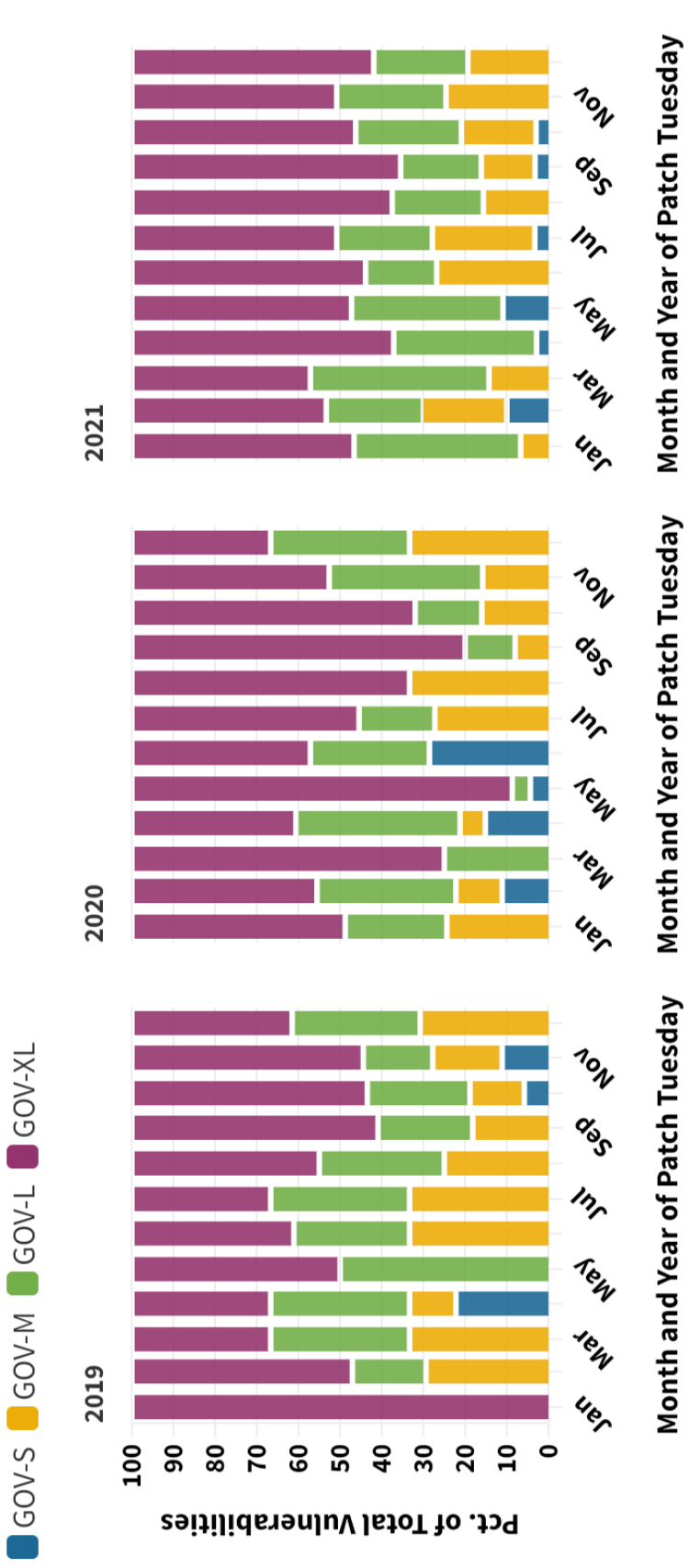


Figure 63: Vulnerabilities by month and year of Patch Tuesday For CVE-IDs between 2019 and 2021 for government facilities sector.

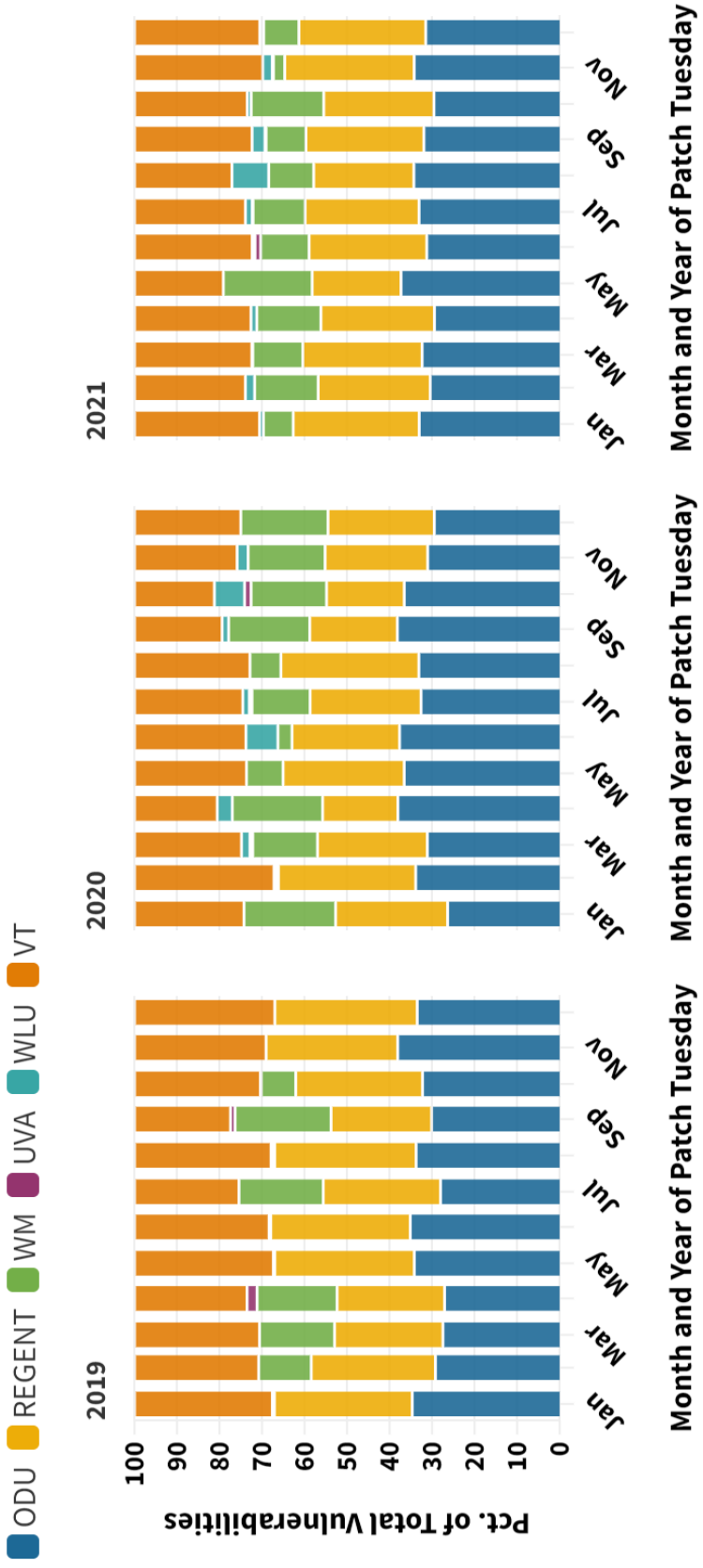


Figure 64: Vulnerabilities by month and year of Patch Tuesday release of CVE-IDs between 2019 and 2021 for the education subsector.

7.2 NORMALIZED DISCOUNTED CUMULATIVE GAIN

Ranking vulnerabilities in terms of descending relevance would make an IT administrators life much easier when determining which CVE-IDs to patch. They would then only need to focus their attention on the most relevant vulnerabilities located towards the top of the list and disregard lower ranked entries. However, solving the task of automatically ranking vulnerabilities is non-trivial and can be subjective. As discussed in Chapter 2 and shown in Figures 63 and 64, the space of relevant vulnerabilities is unstructured and increases over time.

The field of cybersecurity itself has not been quantified in a standard way so as to allow a consensus approach for measuring, testing, and comparing the accuracy of a ranking model. Therefore, this research, similar to others in related work, will build upon measurements derived from the field of IR, specifically metrics that might be applied to probabilistic (i.e., similarities) or feature-based retrieval models. Evaluation measures for an IR system are used to assess how well the search results satisfied the user's query intent. We leveraged an approach found in recommender systems which use the Normalized Discounted Cumulative Gain (nDCG) [87] score to evaluate the ranking of items (e.g., individual vulnerabilities) in a collection (e.g., NVD). The nDCG score varies from 0.0 to 1.0, with 1.0 representing the ideal ranking order. The nDCG is commonly used to evaluate Search Engine Result Pages (SERPs) where the position of an entry, not just on the first page but in comparison to other entries on the page (i.e., towards the top), indicates the most relevant search results. The higher ranked pages are more likely to gain the attention of the consumer. In this research, we apply the same approach towards creating the "to do" list for patching vulnerabilities. The order is important to ensure higher ranked CVE-IDs are considered first. The main difficulty encountered when using nDCG is the availability of an ideal ordering of results when feedback (e.g., human judgment) is not available. We addressed this shortcoming using a data-driven proxy. We introduced Policy Four in Section 6.2 which itself applies threat-centric criteria but also considers whether a CVE-ID has already been exploited during its lifetime.

In order to compare the results of rankings between each relevance policy and the ideal ranking (Policy Four), we calculated the nDCG of each CVE-ID for every organizational interaction with our ranking system. The nDCG values were averaged for each weekly collection of CVE-IDs to obtain a measure of the average performance of our ranking algorithms. The application of nDCG in this study was interpreted as follows:

1. "G" is for gain – corresponds to the magnitude of each vulnerability's relevance.

2. "C" is for cumulative – refers to the cumulative gain, or summed total, of every vulnerability's relevance score.

3. "D" is for discounted – divides each vulnerability's scored relevance by the scored relevance of the associated ideal policy to reflect our goal of having the most relevant vulnerabilities ranked towards the top of our mitigation lists.

4. "N" is for normalized – divides Discounted Cumulative Gain (DCG) scores by ideal DCG scores calculated for a ground truth data set, as represented by the relevance scores and ranking resulting from our ideal policies which were personalized with foreknowledge of exploited vulnerabilities contained within historical ExploitDB and CISA KEV intrusion detection reports.

Once the relevance value is computed for each CVE-ID using the algorithms in Section 6.2, we proceeded to rank each entry based on the relevance value and computed the nDCG using the formulas shown below.

$$DCG_k = \sum_{i=1}^{k} \frac{2^{rel_i} - 1}{\log_2 (i + 1)} \tag{1}$$

The cumulative gain at K is the sum of gains of the first K items recommended. $iDCG_k$ is the maximum possible (ideal) DCG for a given set of queries, vulnerabilities, and relevance scores.

$$nDCG_k = \frac{DCG_k}{iDCG_k} \tag{2}$$

The chart in Figure 65 illustrates the average values of nDCG for each position K based on weekly vulnerability collections. The number of observations ranges from 383 when K=1 down to 16 when K=100. The x-axis reports the rank (from 1 to 100), while the y-axis displays the respective value of nDCG@K. Analyzing the obtained results, we observe the CVSS Base Score performs moderately well when K=1, decreases as K extends from 5 to 50 CVE-IDs, then returns to moderate performance when K=50. The maximum efficiency using the CVSS Base Score appears to occur at opposite ends of the spectrum (K=1, K=100), either a minimal number or alot of CVE-IDs to remediate. The APT Threat policy does not appear to be impacted by the number of weekly CVE-IDs. This ranking policy performs at a consistent level regardless of the number of CVE-IDs encountered.

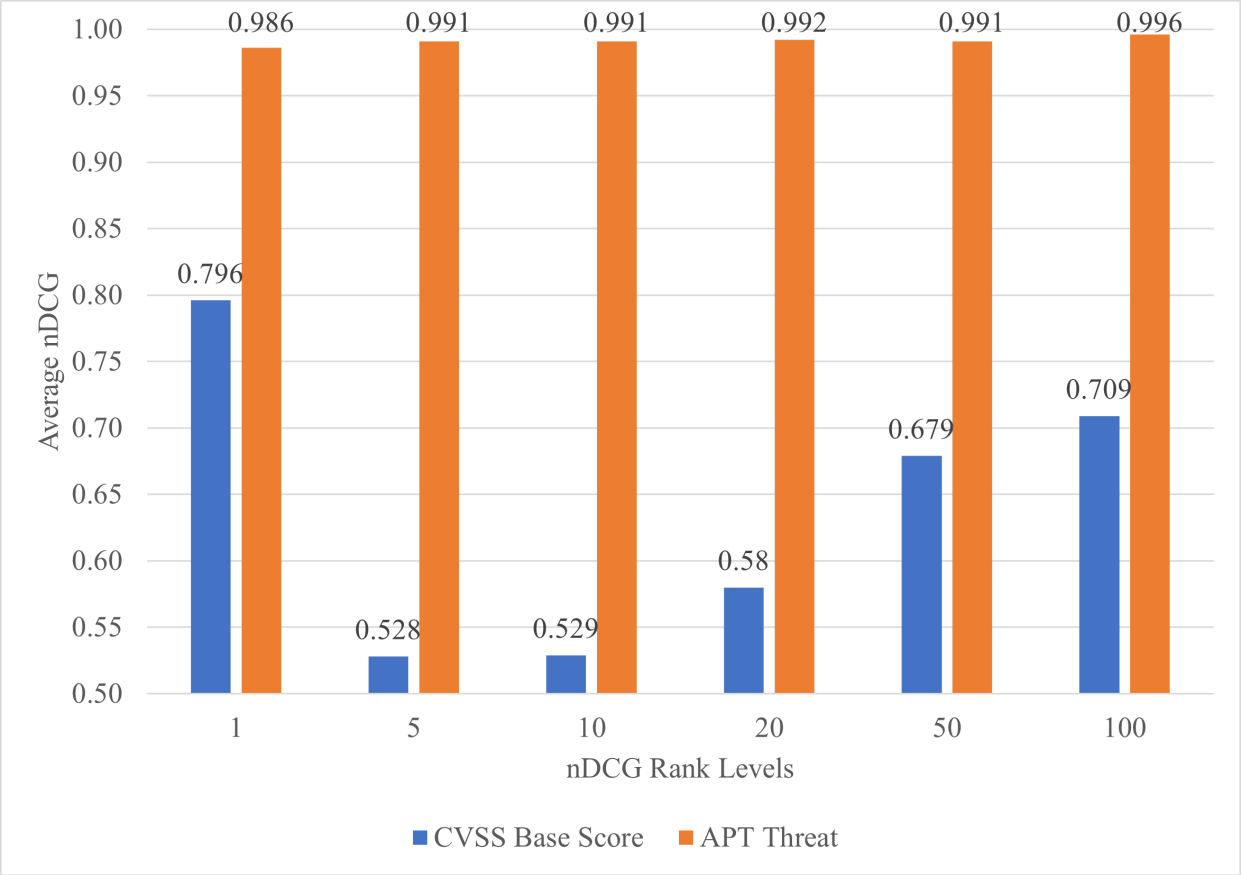


Figure 65: Average value of nDCG at different rank levels (K) for CVSS Base Score versus APT Threat policy for the ODU, REGENT, and WM organizations.

7.3 TESTING AND EVALUATING THE POLICIES

The number of applicable CVE-IDs to be evaluated each week can vary for each organization. Therefore, we need to normalize the cumulative gain at each ranking position for a chosen number of vulnerabilities to calculate nDCG. Using the average, weekly vulnerability traffic across all organizations as shown in Tables 40 and 42 as general guidance, we established a threshold of 20 assigned CVE-IDs during a given week as the minimum number needed to apply a relevance ranking policy. Only the GOV-XL, ODU, REGENT, UVA, WLU, and WM organizations consistently met this threshold. We excluded GOV-XL, UVA, and WLU from further examination in this section as there were numerous weeks

where no published CVE-IDs were applicable to the organization's installed software. This resulted in a small number of weekly collections for these universities, 3 weeks, 14 weeks, and 19 weeks respectively, across all three years of this study. For the remaining organizations with more than 50 weekly observations, ODU, REGENT, and WM, we collected the necessary features using the cyber intelligence data sources identified in Section 2 to compute a relevance score, rank the CVE-IDs, and calculate nDCG using Policy Four as the ideal ranking. Based on the policy descriptions, only the CVSS V3.1 base score was needed to evaluate Policy One. For all ranking policies, the set of applicable CVE-IDs were ranked in descending order by relevance score, then subsequently ordered by CVE-ID to avoid ties. We evaluated the performance of Policy One (CVSS Base Score) against our threat-centric Policies Two (APT Threat) and Three (General Threat). Finally, we determined the patch cost (in non-monetary units) for the top 20 CVE-IDs based on the CVSS scoring approach defined by Fhurwith et al. [61] where Low=0.25, Medium=1, High=1.50, and Critical=3.00.

7.3.1 MEASURING RANKING QUALITY

For the two threat-centric ranking policies we defined, we measured the average performance across all three years of this study using nDCG@20. China was chosen as the APT group of interest (Section 5.4.1, Table 18) for vulnerabilities impacting ODU, REGENT, and WM. For the cumulative gain evaluations, we varied the parameters in Algorithm 1, APT Threat Relevance Scoring, as follows:

- Graph queries were used to associate an organization with its country of residence and associated DHS sector.
- The attack group country of interest was verified against the State Department's List of Independent States (Section 5.3).
- Risk appetite was used as a predictive measure to correlate with the EPSS percentile for probability of exploit and organization's acceptance or risk. If the risk appetite is not available, 88% is used as the default setting (Section 2.3.3).
- The week start date was set to coincide with the corresponding Patch Tuesday.
- Each feature was assessed using a binary weighting [0, 1] and aggregated to determine the final relevance and ideal score.

- Each CVE-ID must be applicable to organization's installed software to be included in the candidate vulnerability list. This assumption results in the value 1 being the minimum, assigned relevance score.
- For Policy One (CVSS Base Score), the CVE-IDs were sorted in descending order from [10, 0] to determine the ranking.
- For Policy Two (APT Threat) and Policy Three (General Threat), the CVE-IDs were sorted in descending order based on the relevance score (i.e., most relevant CVE-IDs presented first).
- The DCG was calculated for the CVSS Base Score and Threat policies, then compared to the Policy Four (Ideal) to determine nDCG.

As defined in Section 7.2, the nDCG is measured on a scale of 0.0 to 1.0, with 1.0 indicating the ideal ranking order has been achieved. Our goal is to obtain a nDCG score close to 1 for each threat policy. There were three instances where an nDCG@20 of 1 was associated with a known exploit. We observed 39 CVE-IDs over 98 weeks of collection had a relevance score of 6 or higher (close to ideal relevance score), were known to have been exploited, and resulted in a nDCG@20 of 0.9 or higher.

Table 43 shows the average nDCG@20 for each organization. The average nDCG@20 of 0.99 indicates our threat-centric ranking scheme performs better than the CVSS Base Score approach. The average difference in nDCG@20 of 0.41 indicates the APT Threat policy performs 71.5% better than the CVSS base score as an indicator of vulnerabilities that might be targeted by an APT group.

Table 43: Average performance of Policy One (CVSS Base Score) versus Policy Two (APT Threat) where China is the source region of interest (nDCG@20).

Year	CVSS Base Score	APT Threat China	Avg. Diff in nDCG	Known Exploits
		ODU		
2019	0.601	0.996	0.394	4
2020	0.557	0.998	0.441	2
2021	0.571	0.986	0.415	12
		REGENT		
2019	0.592	0.999	0.407	2
2020	0.557	0.998	0.441	1
2021	0.585	0.985	0.399	12
		WM		
2019	0.598	0.998	0.400	3
2020	0.565	0.998	0.433	1
2021	0.585	0.985	0.399	12

In our results, the nDCG@20 measures for the CVSS Base Score policy were in the range of [0.343, 1], as shown in Figure 66. Lower values for nDCG@20 were observed with the CVSS Base Score ranking when the number of vulnerabilities collected exceeded the minimum threshold by more than 1000% (e.g., 200+). Higher nDCG@20 values were observed when the number of vulnerabilities was closer to the threshold (e.g., 20 to 30). The APT Threat ranking was minimally impacted by the number of vulnerabilities and ranged from [0.878, 1].

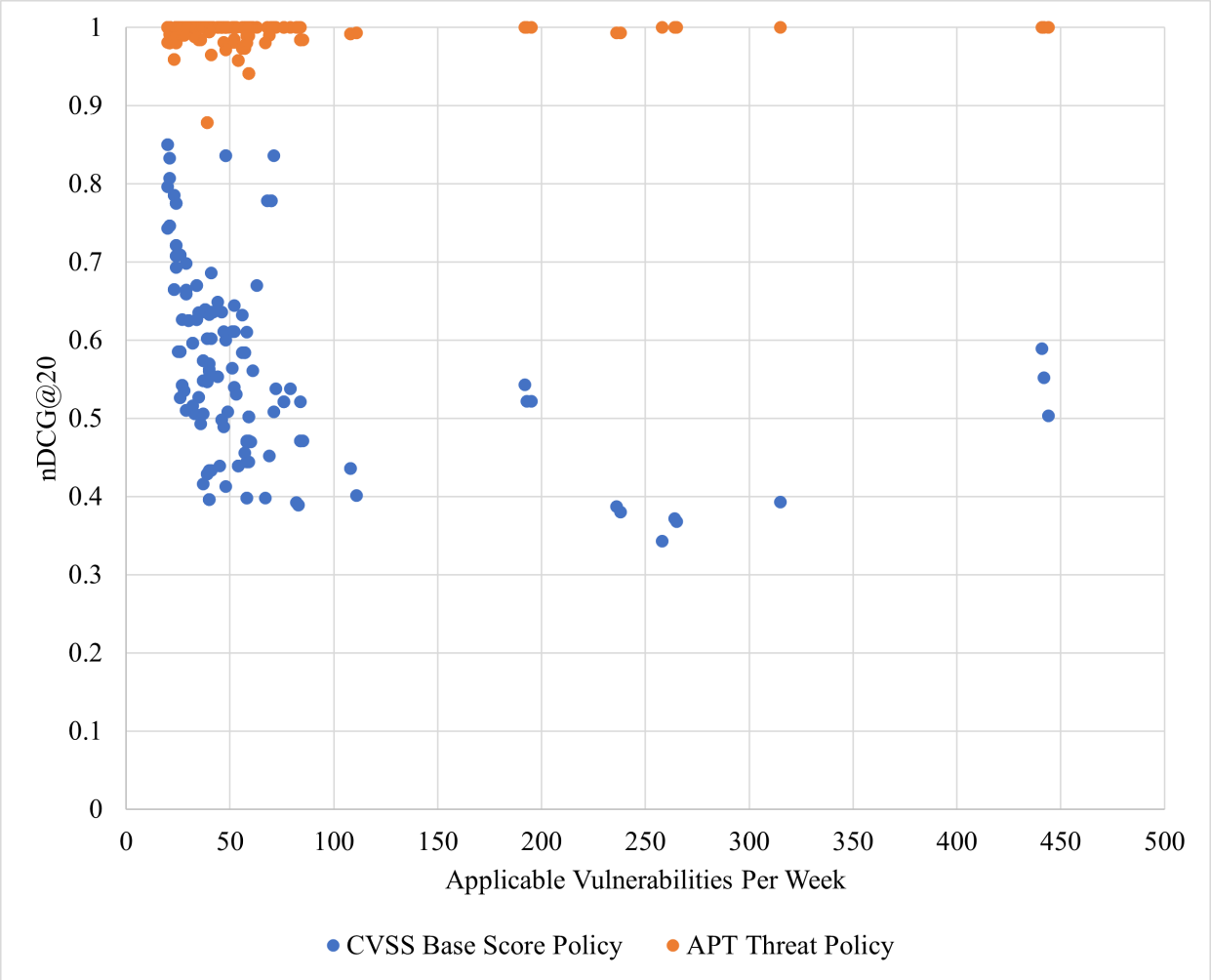


Figure 66: nDCG@20 for CVSS Base Score versus APT Threat policy for the ODU, RE-GENT, and WM organizations.

Using all of the weekly observations (n=163) across organizations, we performed a paired t-test [173] to compare the mean of the nDCG for the CVSS Base Score policy against the APT Threat ranking policy. The null hypothesis asserts there is no difference in the recommended ordering of CVE-IDs between the ranking policies. Our alternative hypothesis is that the APT Threat policy rankings are significantly closer to the ideal ranking (Policy Four) than the CVSS Base Score rankings. Results of the paired t-test indicated that there is a significant large difference between CVSS Base Score (Mean = 0.58, STDEV = 0.1) and APT Threat (Mean = 0.992, STDEV = 0.02) and the p-value equals 0. The APT

Threat population's nDCG@20 average is considered to be greater than the CVSS Base Score population's average, and the difference is large enough to be statistically significant. A histogram depicting the frequency of differences in the nDCG@20 between policies is shown in Figure 67.

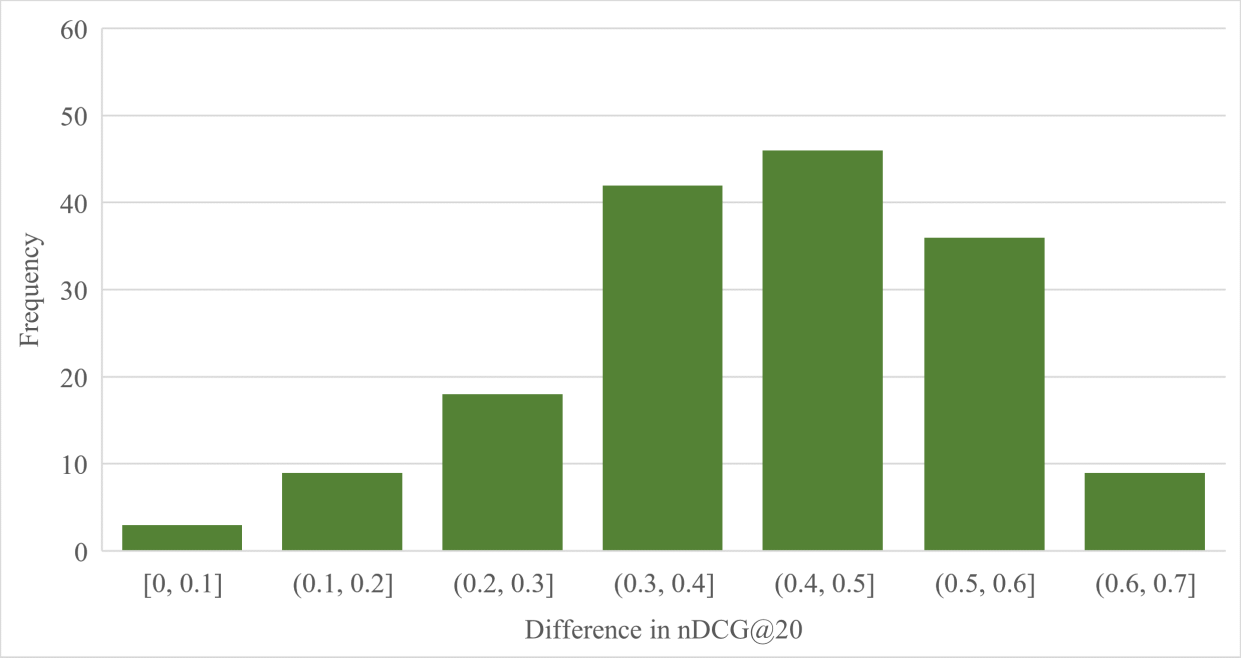


Figure 67: Difference in nDCG@20 across observations between the CVSS Base Score and APT Threat policy for the ODU, REGENT, and WM organizations.

Similar overall results were observed for the General Threat policy as shown in Table 44. The average difference in nDCG@20 of 0.35 indicates the Generalized Threat policy performs 91.3% better than the CVSS Base Score as an indicator of vulnerabilities that might be targeted by any highly-skilled cyber threat actor.

Table 44: Average performance of Policy One (CVSS Base Score) versus Policy Three (General Threat) with a highly skilled adversary (nDCG@20).

Year	CVSS Base Score	General Threat Highly Skilled	Avg. Diff in nDCG	Known Exploits
		ODU		
2019	0.543	0.988	0.444	4
2020	0.548	0.998	0.450	2
2021	0.474	0.986	0.511	12
		REGENT		
2019	0.528	0.995	0.467	2
2020	0.512	0.999	0.487	1
2021	0.500	0.984	0.484	12
		WM		
2019	0.538	0.992	0.454	3
2020	0.520	0.999	0.478	1
2021	0.499	0.984	0.485	12

The nDCG@20 measures for the CVSS Base Score Policy in the range of [0.278, 0.848], as shown in Figure 68, were slightly lower than what we observed for the APT Threat policy. While the General Threat policy is not specific to a named APT group, we observed 21 of 49 CVE-IDs known to have been exploited received the maximum relevance score (7) allowed under this policy.

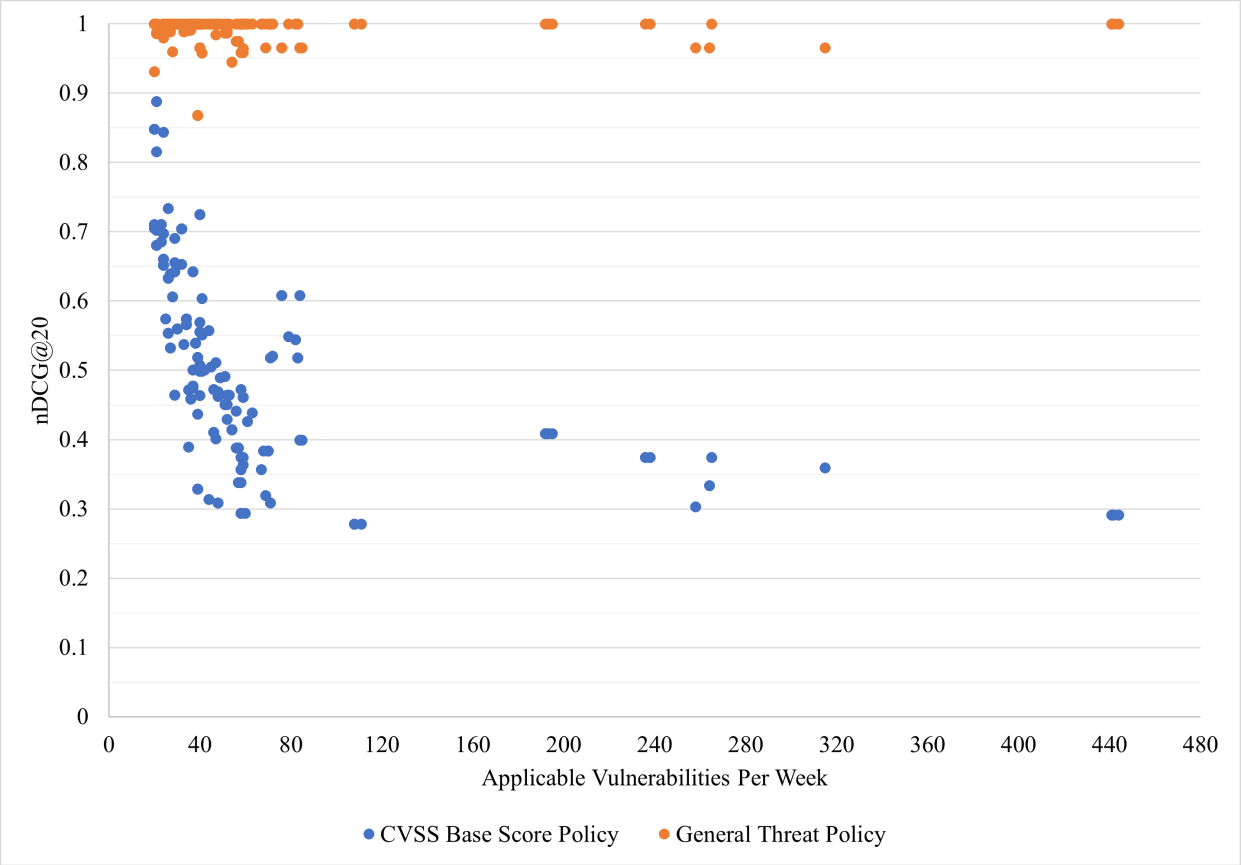


Figure 68: nDCG@20 for the CVSS Base Score versus the General Threat policy for the ODU, REGENT, and WM organizations.

Using all of the weekly observations (n=163) across organizations, we performed a paired t-test [173] to compare the mean of the nDCG for the CVSS Base Score policy against the General Threat ranking policy. The null hypothesis asserts there is no difference in the recommended ordering of CVE-IDs between the ranking policies. Our alternative hypothesis is that the General Threat policy rankings are significantly closer to the ideal ranking (Policy Four) than the CVSS Base Score rankings. Results of the paired t-test indicated that there is a significant large difference between CVSS Base Score (Mean = 0.512, STDEV = 0.139) and General Threat (Mean = 0.99, STDEV = 0.022) and the p-value equals 0. The General Threat population's average nDCG@20 is considered to be greater than the CVSS Base Score population's average, and the difference is large enough to be statistically significant. A histogram of the difference in the nDCG@20 between policies is shown in Figure 69.

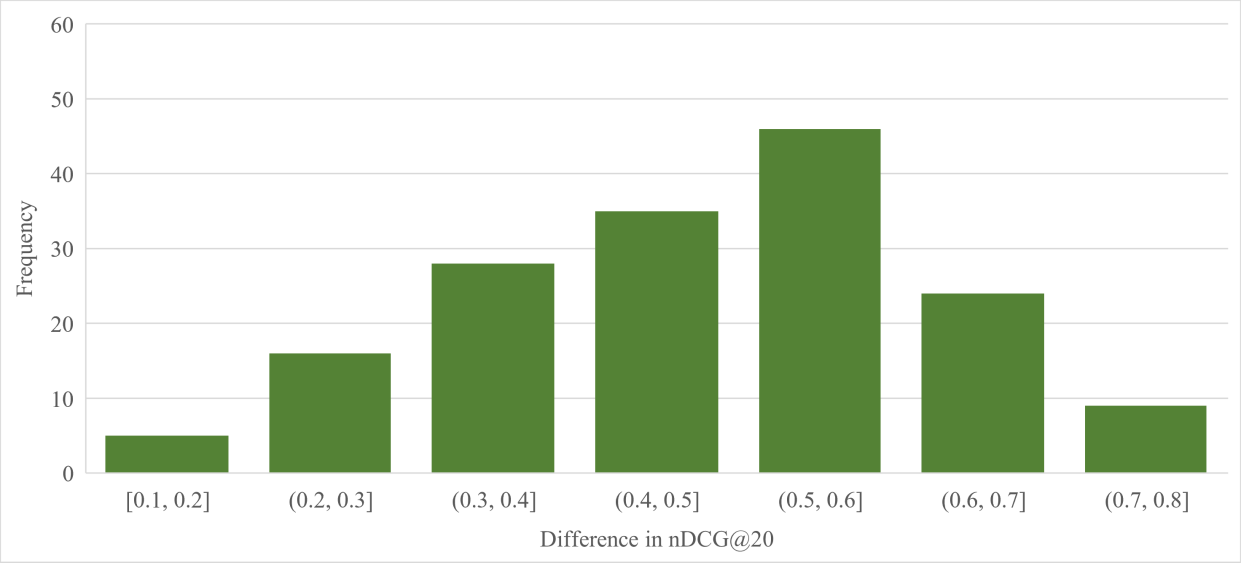


Figure 69: Difference in nDCG@20 across observations between the CVSS Base Score and General Threat policy for ODU, REGENT, and WM organizations.

These results confirm our understanding that CVSS base score metrics do not contain a data element or scoring component that allows for enumeration of a specific threat. The paired t-test indicates that the difference in the recommended ranking positions of CVE-IDs between policies is statistically significant (p-value equals 0). Therefore, any relevance ranking based solely on the CVSS base score will fall short of the organization's specified goals. These results also provide another indication that the severity of a vulnerability, as measured by its CVSS base score, may not be the optimal ranking approach for every organization.

7.3.2 COST OF PATCH PRIORITIZATION

Vulnerability remediation has some fundamental truths. First, there are too many vulnerabilities to fix them all immediately. Past research has shown that organizations are able to fix between 5% and 20% [57] of known vulnerabilities per month. In this section, we examined the annualized cost of remediating the top 20 vulnerabilities as ranked by either the CVSS Base Score or our APT and General Threat policies. The non-monetary units, defined by Fhurwith et al. (Section 3.2.1), associated with patching are shown in Table 45.

In all cases, we observed decreases in the range of 23.3% when the APT Threat policy was used for prioritizing CVE-IDs for remediation. Specifically, decreases of 498 units for ODU, 390.5 units for REGENT, and 455.75 units for WM were realized over the three year period noted.

Table 45: Difference in the cost of patching the top 20 CVE-IDs for Policy One (CVSS Base Score) versus Policy Two (APT Threat) where China is the source region of interest.

Year	CVSS Base Score Cost	APT Threat China Cost	Average Savings
		ODU	
2019	631.50	449.25	182.25
2020	531.00	439.00	92.00
2021	994.50	770.00	224.50
		REGENT	
2019	604.75	422.25	182.50
2020	375.50	308.50	67.00
2021	960.00	752.00	208.00
		WM	
2019	603.75	424.75	179.00
2020	374.00	308.50	65.50
2021	960.00	748.75	211.25

Using all of the weekly observations (n=163) across organizations, we performed a paired t-test [173] to compare the mean of the patch costs for the CVSS Base Score policy against the APT Threat ranking policy. The null hypothesis asserts there is no difference in patch costs based on the recommended ordering of CVE-IDs between the ranking policies. Our alternative hypothesis is that the APT Threat policy's cost of patching is significantly lower than the CVSS Base Score costs. Results of the paired t-test indicated that there is a

significant large difference between CVSS Base Score (Mean = 37.025, STDEV = 10.291) and APT Threat (Mean = 28.362, STDEV = 5.475) and the p-value equals 7.45e-27. The APT Threat population's average patch cost is considered to be less than the CVSS Base Score population's average, and the difference is large enough to be statistically significant. A histogram depicting the frequency of differences in the patch costs between policies is shown in Figure 70.

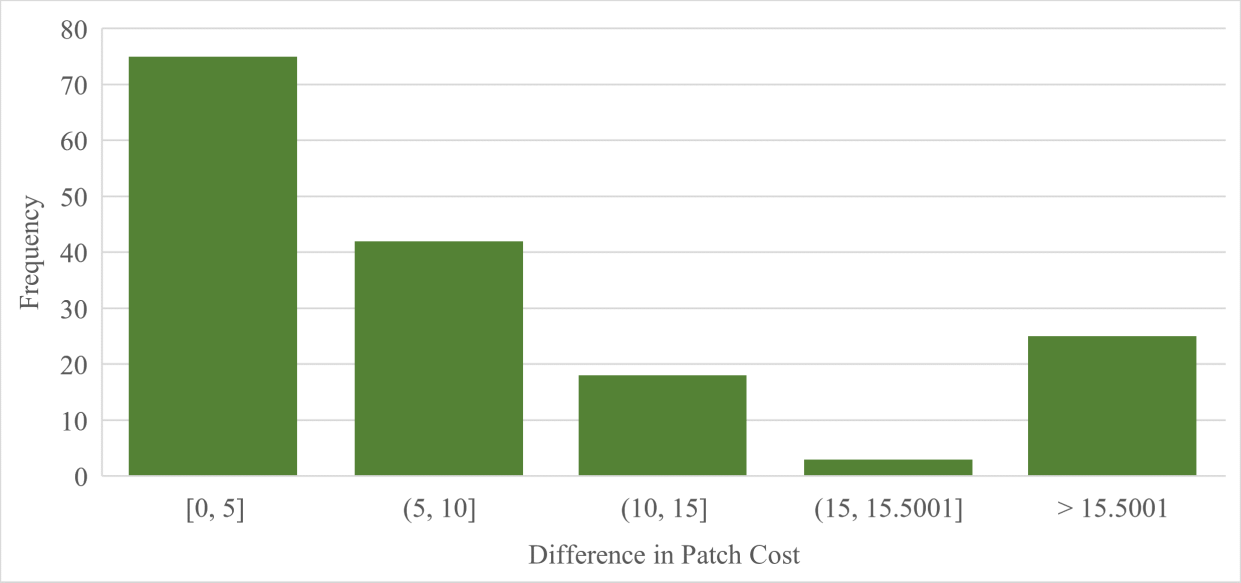


Figure 70: Difference in patch costs across weekly observations between the CVSS Base Score and APT Threat policy for the ODU, REGENT, and WM organizations.

We observed increased savings in patch costs using the General Threat policy as shown in Table 46. The cost of patching remains the same across all organizations using the CVSS Base Score. However, for each organization we see additional savings over those already noted using the APT Threat policy (Table 45). Specifically, decreases of 548.25 units for ODU, 500.75 units for REGENT, and 499.75 for WM represent an average 25.6% improvement over the CVSS Base Score approach.

Table 46: Difference in the cost of patching the top 20 CVE-IDs for Policy One (CVSS Base Score) versus Policy Three (General Threat) from a highly skilled adversary.

Year	CVSS Base Score Cost	General Threat Cost	Average Savings
		ODU	
2019	631.50	438.50	193.00
2020	531.00	424.50	106.50
2021	994.50	745.75	248.75
		REGENT	
2019	604.75	412.00	192.75
2020	375.50	294.50	81.00
2021	960.00	733.00	227.00
		WM	
2019	603.75	412.50	191.25
2020	374.00	296.50	77.50
2021	960.00	729.00	231.00

Using all of the weekly observations (n=163) across organizations, we performed a paired t-test [173] to compare the mean of the patch costs for the CVSS Base Score policy against the General Threat ranking policy. The null hypothesis asserts there is no difference in patch costs based on the recommended ordering of CVE-IDs between the ranking policies. Our alternative hypothesis is that the General Threat policy's cost of patching is significantly lower than the CVSS Base Score costs. Results of the paired t-test indicated that there is a significant large difference between CVSS Base Score (Mean = 37.025, STDEV = 10.291) and APT Threat (Mean = 27.523, STDEV = 4.905) and the p-value equals 9.989e-30. The APT Threat population's average patch cost is considered to be less than the CVSS Base Score population's average, and the difference is large enough to be statistically significant. A histogram depicting the frequency of differences in the patch costs between policies is shown in Figure 71.

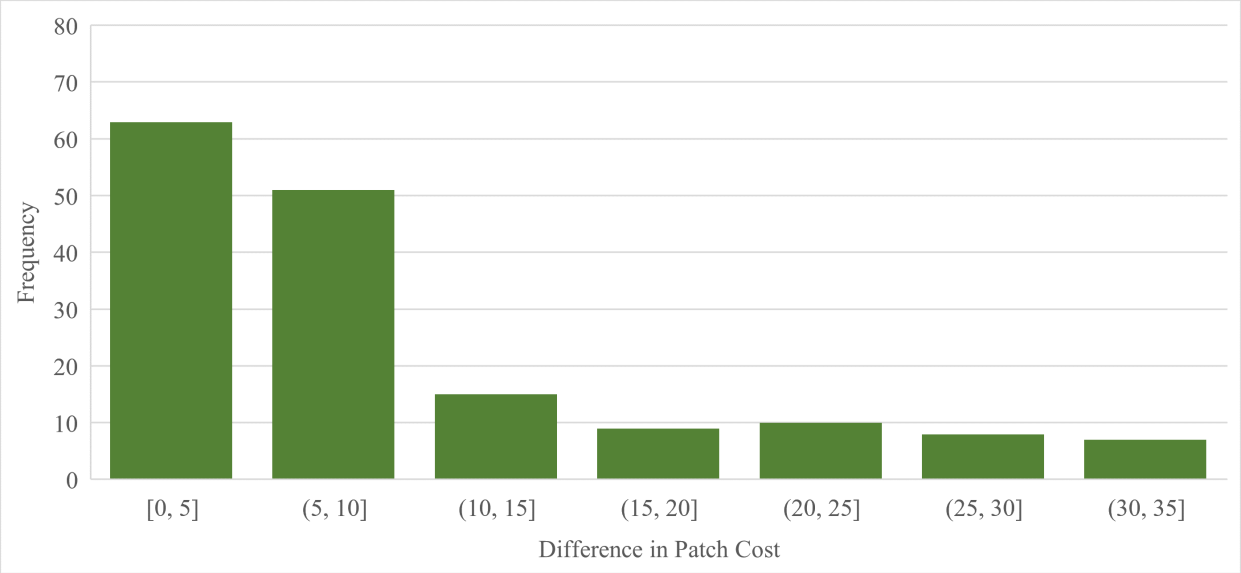


Figure 71: Histogram of the difference in patch costs across weekly observations between the CVSS Base Score and General Threat policy for the ODU, REGENT, and WM organizations.

7.3.3 PREDICTING EXPLOITS

According to the FIRST organization, only a small subset (2%-7%) of published vulnerabilities are ever seen to be exploited in the wild [57]. Given that such a small number of CVE-IDs are actually exploited, it is advantageous for organizations to leverage as much insight as possible to identify potential threats. In this section, we demonstrate how the APT Threat ranking policy was used to prioritize a vulnerability with a known exploit. The ODU organization identified 39 CVE-IDs to be mitigated during the week of 23-November-2021. In this case study, the top 20 were ranked in accordance with our APT Threat policy as shown in Table 47. We note that three CVE-IDs in this group, CVE-2021-38000, CVE-2021-30632, and CVE-2021-30633, have known exploits. The CISA Known Exploits entry for CVE-2021-38000, which impacts Google Chrome, is shown in Figure 72. The entries in Table 47 show that all three CVE-IDs would have been identified as relevant using our ranking policy. However, CVE-2021-38000 would have been ranked at position 29 based on its CVSS base score of 6.1 (medium severity) and would have fallen outside of the top 20

range for remediation by IT administrators at ODU. The APT Threat policy elevated this CVE-ID to position number 3 based on the high relevance score assigned. Even though the top three CVE-IDs have the same relevance score, CVE-2021-38000 is in the third position based on the secondary ranking criteria we applied to avoid ties.

Table 47: Application of ranking policies by ODU for vulnerabilities published during the week of 23-November-2021.

CVE-ID	**CVSS Base Score**	**Relevance Score**	**CVSS Base Score Rank**	**APT Threat China Rank**	**Known Exploit**
CVE-2021-37966	4.3	6	34	1	
CVE-2021-37999	6.1	6	28	2	
CVE-2021-38000	**6.1**	**6**	**29**	**3**	**Yes**
CVE-2021-30542	8.8	2	5	4	
CVE-2021-30543	8.8	2	6	5	
CVE-2021-30626	8.8	2	7	6	
CVE-2021-30627	8.8	2	8	7	
CVE-2021-30628	8.8	2	9	8	
CVE-2021-30629	8.8	2	10	9	
CVE-2021-30630	4.3	2	31	10	
CVE-2021-30632	**8.8**	**2**	**11**	**11**	**Yes**
CVE-2021-30633	**9.6**	**2**	**2**	**12**	**Yes**
CVE-2021-34423	9.8	2	1	13	
CVE-2021-34424	7.5	2	26	14	
CVE-2021-37956	8.8	2	12	15	
CVE-2021-37957	8.8	2	13	16	
CVE-2021-37958	5.4	2	30	17	
CVE-2021-37959	8.8	2	14	18	
CVE-2021-37961	8.8	2	15	19	
CVE-2021-37962	8.8	2	16	20	

We also used this study as an opportunity to revisit the limitations expressed by Tatem et. al [151] and discussed in Section 5.4.3. Using CVE-2021-38000 as the example vulnerability and ODU as the organization employing a ranking policy, Figure 73 demonstrates our ability, via graph queries, to create a complete solution to correlate attackers, exploits, and vulnerabilities in a single model. The legend for the node labels and relationships was defined in Chapter 5, Table 15.

7.4 KNOWN LIMITATIONS

The following provides a description of known limitations the application of our framework, many of which are constraints imposed by the cyber intelligence used for analysis and the manner in which this data is collected by data owners.

- Organizations must have a methodology to accurately construct a software inventory that can be correlated with an entry in the Common Platform Enumeration (CPE) database. Vulnerabilities cannot be allocated without a CPE-ID and low fidelity inventory reporting may result in residual cyber risk. The relevance ranking policies we have identified can only be effectively applied to a known software architecture.

- The attack group list in MITRE ATT&CK is not all encompassing. A Google search will identify emerging APT groups that are not included in the MITRE's enterprise matrices.

- The source for proof of concept code entries collected via ExploitDB does not include a time component which indicates when the POC entry was made. As a result, it is not possible to discretely link the CVE-ID's publication or modification date with the subsequent appearance of an intrusion report. The inclusion of a timestamp would have allowed us to evaluate the predictive portion of our policies based on a timeline of events. Our approach is naive with regard to exploitation and does not consider the publication date for exploit code maturity using ExploitDB. The ExploitDB to CVE mapping webpage is also not well covered in the Internet Archive.

- The EPSS probability scores and percentiles are dynamic and should be collected near the time of the CVE publication date. In order to maintain consistency in our data set, all cyber intelligence data was collected and frozen for analysis as of 31-December-2021. Future work can utilize the API provided by the EPSS team to dynamically collect the scores and percentiles in real time. This is a candidate for future work.

CVE	Vendor/Project	Product	Vulnerability Name	Date Added to Catalog	Short Description	Action	Due Date
CVE-2021-38000	Google	Chromium V8 Engine	Google Chromium V8 Insufficient Input Validation Vulnerability	2021-11-03		Apply updates per vendor instructions.	2021-11-17

Figure 72: CISA known exploits catalog entry for CVE-2021-38000 (Reproduced from [40]).

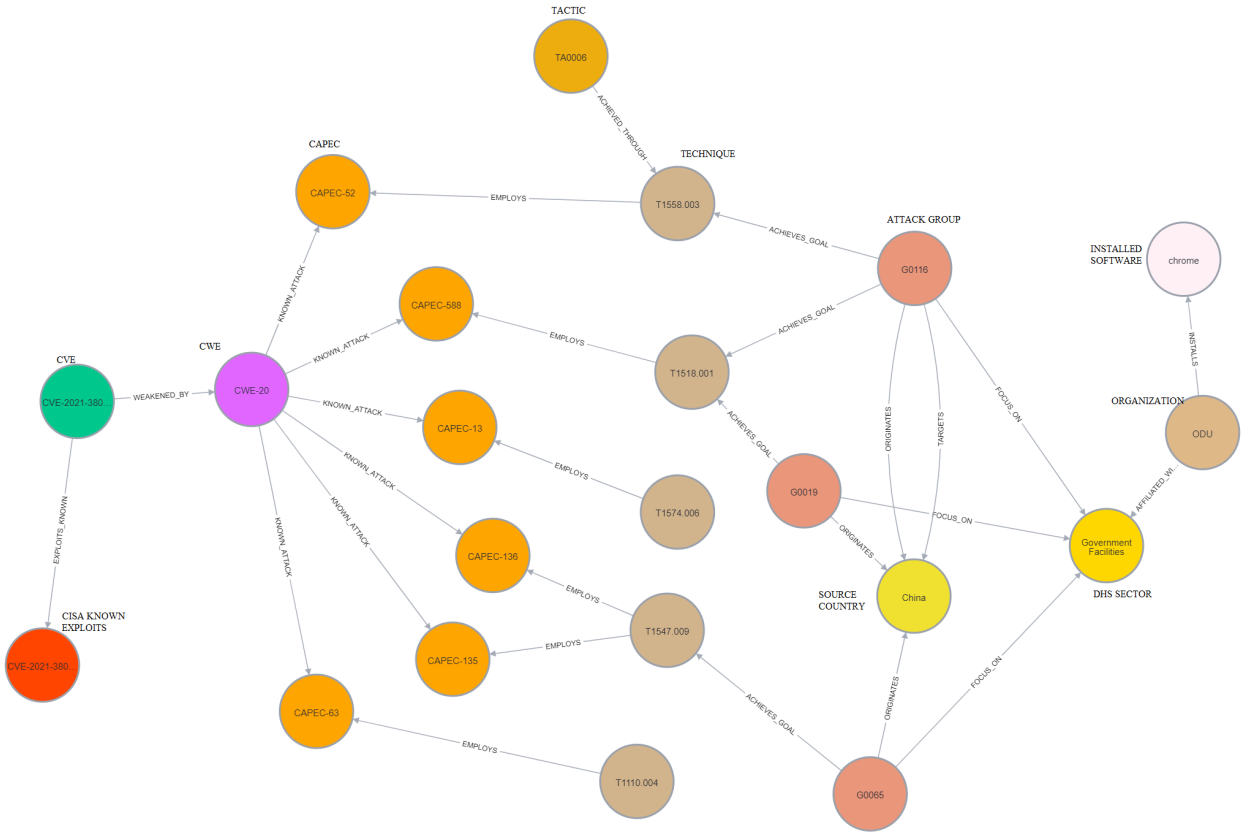


Figure 73: Graph query shows the traceability from APT groups (dark orange nodes) operating in China (yellow node) to the exploit of CVE-2021-38000 (red node) based on the techniques employed (brown nodes).

- The optimum ordering approach that we prescribe using intentional ranking policies may not ease patch hesitancy (Section 1.1.1) or prevent a culture of "wait and see" with regard to patching vulnerabilities. Our policies also cannot control the quality of vendor patch distributions on Patch Tuesday that in some cases lead to Recall Thursday. These scenarios are outside the scope of this research. Our ranking policies can, however, reduce the amount of unnecessary work spent patching CVE-IDs that are neither applicable nor associated with a known cyber threat actor.

7.5 CHAPTER SUMMARY

In this chapter we provided two operational scenarios to evaluate the threat-centric ranking policies defined in Section 6.2. We used these scenarios to demonstrate that severity of a vulnerability may not be the optimal ranking approach for every organization. We used nDCG to evaluate the performance of each policy when compared to an industry standard CVSS Base Score approach. We used annualized performance to recognize a 71.6% to 91.3% improvement in rankings when a personalized approach is implemented. Further, we demonstrated via an analysis of patch costs that significant decreases, on the order of 23.3% to 25.5%, can realized using our threat policies as well. We emphasized how the relevance scores associated with a specific threat policy could help to prioritize vulnerabilities with greater likelihood to be exploited. In Section 7.3.3, we illustrated via graph queries and relationships how the standardized cyber intelligence data set we curated for this study could be used to address known challenges cited in recent research. Finally, we identified known limitations that should be considered when implementing the framework we have defined.

Chapter 8

CONTRIBUTIONS, FUTURE WORK, AND CONCLUSIONS

In this chapter we provide a review of our research questions and how each was addressed in this dissertation. In Section 8.1 we enumerate our contributions introducing and further exploring the nuances of the threat-centric relevance ranking framework. Section 8.2 describes future work beyond the scope but facilitated by this dissertation. In Section 8.3, we summarize our conclusions in exploring the research described in this dissertation. The remainder of this section details how we addressed the research questions.

RQ1: What are the factors that can be used to model attack vectors and security threats based on the skill level of a cyber adversary and their motivation to target a specific industry domain (e.g., national defense, higher education, finance, health care)?

Our goal was to define a repeatable approach to link disparate cyber intelligence data sets in a way that would allow efficient queries of adversary tactics and identification of specific targets. In order to achieve this goal, we applied information retrieval and data mining techniques to collect and curate information from the CVE, CWE, CAPEC, MITRE ATT&CK, ExploitDB, and CISA data sets. Chapter 4 describes the methods we employed to extract pertinent metadata and create the schema for an initial knowledge graph.

In Chapter 5, we described supplemental data we curated related to make the attack group data found in MITRE ATT&CK more actionable. We performed data mining on adversary groups descriptions, then used natural language processing to extract keywords to determine the country from which the group operates. We analyzed the group descriptions to determine the most prevalent country of origin as shown in Chapter 5.

Finally, we created a novel approach to link attacker techniques to vulnerabilities (Section 5.4.3) by again extracting keyword phrases from attack group data to extract DHS sectors that might be targeted and provide a methodology to link sectors and countries to a specific organization. We found Government Facilities are highly ranked targets for APT groups with the top three adversarial threats originating from China, Russia, and Iran. We used a known exploit related to a specific CVE-ID to provide practical example in Chapter 7 that illustrates how the graph schema we curated could be used to create a complete solution to correlate attackers, exploits, and vulnerabilities in a single model.

RQ2: What are the characteristics needed to define vulnerability ranking policies that improve the return on investment (ROI) of applied mitigations, compared to traditional CVSS Base Score policies, relevant to the organization's specific mitigation goals and priorities?

Our goal was to define an approach to cybersecurity vulnerability mitigation which improves upon rankings that employ strategies based on the global CVSS metrics associated with known software vulnerabilities published in the NVD. We focused on types of attacks and attackers to create two ranking policies, Section 6.2, that could be powered by the data sets in our study to facilitate a data-driven approach for ranking CVE-IDs as they are published weekly. We used techniques from information retrieval to the rank the quality of the threat-centric policies we defined using 13,862 vulnerabilities published between 2019 and 2021. Our evaluation showed a 40% improvement over the CVSS base score for escalating CVE-IDs for mitigation that fit the criteria for tactics and techniques employed by known APT groups. Further, we measured the ROI of patching and realized a similar 40% reduction in annual unit costs.

8.1 CONTRIBUTIONS

This dissertation contributes to the field of cybersecurity vulnerability management by introducing a framework that allows organizations to personalize their mitigation strategies within the context of specific threats from known APT groups or a generalized attacker based on their skill level. Our contributions include the following:

1. We defined a fully transparent and open ranking methodology based on reliable and authoritative sources that allows for due diligence and reference checking (Section 6.2).

2. We described a methodology for standardizing, normalizing, and categorizing cyber intelligence housed in huge volumes of public and government-specific data repositories (Section 4).

3. We developed a knowledge graph schema[1] (Section 5.1) that allows us to identify threat actors who actively target an organization, understand motives and techniques, and track trends relative to the industry and regions where organizations operate. We will leave this analysis for future work. The graph schema is not only useful to understand security risks posed by known adversary behavior, but also for assessing

[1] `https://github.com/correnm/RelevanceRank`

deficiencies and planning improvements to reduce the attack surface which accounts for the specific environment and implemented security controls.

4. We created an approach for characterizing threat actors based on tactics and techniques employed (Section 5.4) that fills the gap identified by Mavroeidis and Hohimer [101].

5. We presented a complete graph-based solution (Section 5.1) that addresses the challenge noted by Tatem et al. [151] to link a threat actor's tactics using the mapping of CVE and CPEs to known exploits.

8.2 FUTURE WORK

Future work includes specific steps to address the known limitations we identified in Section 7.4. This includes applying automation to accelerate the process of calculating and updating relevance scores as new cyber intelligence is published to provide an organization with near real-time rankings.

The Exploit Prediction Scoring System (EPSS) percentiles and probabilities that we used as a predictor of likely exploit is based on every published CVE-ID. It is unlikely that any organization is dealing with every CVE-ID based on their software inventory. The opportunity exists to personalize the percentiles using the subset of vulnerabilities relevant to the organization's network environment. This should increase the accuracy of the predictive quality of future ranking approaches that might be developed. In addition, the EPSS model now includes an API that could be used to query for probabilities on demand vice using the daily download file. The EPSS probability will likely not change, but the relative position (i.e., ranking) of one vulnerability to another will very likely change.

Finally, another source for a leading indicator for ranking and prioritizing vulnerabilities is the CISA Known Exploited Vulnerabilities Catalog. The DHS CISA collects and publishes alerts about current vulnerabilities and exploits as they occur. Embedded within these alerts are specific CVE-IDs and MITRE ATT&CK IDs which in some cases could be mapped back to specific vulnerabilities. Future work could include monitoring and mining the alerts to augment MITRE ATT&CK with recent, real-world information related to activities of known and emerging cyber threat actors and improve the freshness of MITRE's attack group descriptions.

8.3 CONCLUSIONS

In November 2021, CISA issued BOD 22-01, which instructed federal agencies to steer away from focusing solely on CVSS scores and instead to "target vulnerabilities for remediation that have known exploits and are being actively exploited by malicious cyber actors." [42] This BOD was well intentioned, but did not provide an approach or an accessible framework organizations could use to comply. The goal of this research was to demonstrate that aggregating and synthesizing readily accessible, public data sources to provide personalized, automated recommendations that an organization can use to prioritize its vulnerability management strategy will offer significant improvements over what is currently applied using the CVSS base scores. We also sought to define a repeatable approach to link disparate cyber intelligence data sets in a way that would allow efficient queries of adversary tactics and identification of specific DHS sector targets. Once achieved, we showed that vulnerability ranking strategies focused on using the global CVSS base score would be costly and insufficient towards addressing known and emerging cyber threats.

We specifically wanted to increase the cohesion between the plethora of data sets maintained by MITRE, NIST, DHS, and other government R&D agencies. Our work proposes a method to standardize and normalize extracts from attack group descriptions into a knowledge graph schema using Neo4j so we can perform a vulnerability ranking analysis. We employed natural language processing and techniques from the field of information retrieval to allocate attack groups to the country from which their cyber attacks originate. Further, we used data mining to align known tactics and techniques to specific advanced persistent threats. Next, we layered publicly available cyber intelligence related to published vulnerabilities, software weaknesses, attack patterns, and known exploits onto the knowledge graph schema to provide semantic context. Graph queries provided a complete picture of potential and actual threats. Through exploration of the knowledge graph via queries, we were able to allocate the 129 APT groups found in the MITRE ATT&CK enterprise matrix to a country or region, found the United States to be the most targeted country, and obtained traceability from attack groups to vulnerabilities to common weaknesses. The complete graph, as defined in our knowledge graph schema, provides the framework that allows security teams to be very granular in describing and tracking adversarial behavior.

Our work also proposes a method to describe organizational profiles for threat modeling and efficient risk prioritization. We used the characteristics of the CVSS base score, MITRE ATT&CK, and CAPEC to identify the relevance of a set of vulnerabilities as they were published weekly. First, we determine whether the CVE-ID affects the organization's

software inventory. After that, we apply an algorithm, in conjunction with the knowledge graph, how well the vulnerability fits the defined threat model. Next, we score and rank the vulnerabilities based on susceptibility to a type of attack, adversaries of interest, their interest in the organization's DHS sector, and likelihood of exploit. Finally, we measure the annual ROI in patching a minimum set of CVE-IDs each week.

We evaluated our models using a collection of 13,862 CVE-IDs that we allocated across six public and private universities in Virginia and four proxy organizations we defined expressly for the conduct of this study. We observed that smaller organizations may not have consistent exposure to the volume of vulnerabilities to warrant a specific policy. We also observed the same phenomena for our proxy organizations, regardless of size, where the software inventory was intentionally chosen based on rigorous security requirements. For all other organizations, we found that ranking policies based on specific APT and generalized threats consistently outperformed a CVSS base score approach by more the 71%. Further, there was a significant reduction in the annual cost of patching by more than 23%. A paired t-test demonstrated these findings are statistically significant and offer an improvement over the industry standard approach to vulnerability management. These results, in particular, confirm our understanding that CVSS base score metrics do not contain a data element or scoring component that allows for enumeration of a specific threat. Therefore, any relevance ranking based on the CVSS base score will likely fall short of the organization's specified goals. Further, our results provide indication that the severity of a vulnerability may not be the optimal ranking approach for every organization.

Overall, the relevance ranking strategy as described in this study emphasizes the tremendous capability of threat-centric scenarios for ranking and prioritizing vulnerabilities with due consideration of the threat environment. A network defender, who typically has to address thousands of exposed vulnerabilities, is now able to spend fewer resources to patch more vulnerabilities that are much more likely to be exploited and of interest to a specific set of cyber threat actors. The automated data aggregation within the knowledge graph allows for quick checking of new vulnerabilities that affect the most important software and servers. This capability to differentiate among vulnerabilities and how they might be targeted by an adversary has never before been possible.

REFERENCES

[1] Google Trends. https://trends.google.com/trends/?geo=US. Accessed 2021-05-01.

[2] Internet Archive: Wayback Machine. https://archive.org/web/. Accessed 2021-05-01.

[3] National Security Telecommunications and Information Systems Security Policy (NSTISSP) No. 11. https://www.hsdl.org/?view&did=487791, July 2003. Accessed 2022-10-18.

[4] Vulnerability management. https://www.parsolvo.com/vulnerability-management/, 2020. Accessed 2021-02-28.

[5] CVE program report for Q4 calendar year 2020. https://medium.com/@cve_program/cve-program-report-for-q4-calendar-year-2020-4f564146f3b2, January 2021. Accessed 2021-02-28.

[6] C. J. Alberts and A. J. Dorofee. *Managing information security risks: the OCTAVE approach*. Addison-Wesley Professional, 2003.

[7] L. Allodi and S. Etalle. Towards realistic threat modeling: Attack commodification, irrelevant vulnerabilities, and unrealistic assumptions. In *Proceedings of the 2017 Workshop on Automated Decision Making for Active Cyber Defense*, pages 23–26, 2017. doi: 10.1145/3140368.3140372.

[8] L. Allodi and F. Massacci. A preliminary analysis of vulnerability scores for attacks in wild: The EKITS and SYM datasets. In *Proceedings of the 2012 ACM Workshop on Building analysis datasets and gathering experience returns for security*, pages 17–24, 2012.

[9] L. Allodi and F. Massacci. Comparing vulnerability severity and exploits using case-control studies. *ACM Transactions on Information and System Security (TISSEC)*, 17(1):1–20, 2014.

[10] L. Allodi, F. Massacci, and J. Williams. The work-averse cyberattacker model: Theory and evidence from two million attack signatures. *Risk Analysis*, 42(8):1623–1642, 2022.

[11] M. Almukaynizi, E. Nunes, K. Dharaiya, M. Senguttuvan, J. Shakarian, and P. Shakarian. Proactive identification of exploits in the wild through vulnerability mentions online. In *2017 International Conference on Cyber Conflict (CyCon US)*, pages 82–88. IEEE, 2017.

[12] K. Alperin, A. Wollaber, D. Ross, P. Trepagnier, and L. Leonard. Risk prioritization by leveraging latent vulnerability features in a contested environment. In *Proceedings of the 12th ACM Workshop on Artificial Intelligence and Security*, pages 49–57, 2019. doi: 10.1145/3338501.3357365.

[13] A. Anwar, A. Abusnaina, S. Chen, F. Li, and D. Mohaisen. Cleaning the NVD: Comprehensive quality assessment, improvements, and analyses. *arXiv preprint arXiv:2006.15074*, 2020.

[14] Apache Logging Services. Apache Log4j. https://logging.apache.org/log4j/2.x/index.html. Accessed 2022-10-03.

[15] M. Bada, S. Creese, M. Goldsmith, C. Mitchell, and E. Phillips. Computer security incident response teams (CSIRTs): An overview. *The Global Cyber Security Capacity Centre*, 2014.

[16] A. Balapure. Cyber weapon of mass destruction- The Blackhole exploit kit, May 2013. https://resources.infosecinstitute.com/topic/cyber-weapon-of-mass-destruction-the-blackhole-exploit-kit/.

[17] Balbix. CVSS v2 vs CVSS v3. https://www.balbix.com/insights/cvss-v2-vs-cvss-v3/. Accessed 2020-11-20.

[18] S. Barnum. Common Attack Pattern Enumeration and Classification (CAPEC) schema. https://capec.mitre.org/, 2008.

[19] C. Bizer, T. Heath, and T. Berners-Lee. Linked data: The story so far. In *Semantic services, interoperability and web applications: emerging concepts*, pages 205–227. IGI global, 2011.

[20] D. Bradbury. Unveiling the dark web. *Network security*, 2014(4):14–17, 2014.

[21] R. A. Bridges, C. L. Jones, M. D. Iannacone, K. M. Testa, and J. R. Goodall. Automatic labeling for entity extraction in cyber security. *arXiv preprint arXiv:1308.4941*, 2013.

[22] BrightCloud. Project Sonar found 50,000 microsoft DNS servers with a remote code execution vulnerability,... https://twitter.com/BrightCloudLtd/status/1286217139297361920, July 2020. Accessed 2020-11-20.

[23] A. Buttner and N. Ziring. Common Platform Enumeration (CPE)—specification. http://cpe.mitre.org, 2009.

[24] CERT-Bund. Urgent Microsoft update for Windows DNS server... https://twitter.com/certbund/status/1283320290320293888, July 2020. Accessed 2020-11-20.

[25] B. A. Cheikes, K. A. Kent, and D. Waltermire. *Common Platform Enumeration: Naming specification version 2.3*. US Department of Commerce, National Institute of Standards and Technology, 2011.

[26] T. Chen and C. Guestrin. XGBoost: A scalable tree boosting system. In *Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining*, pages 785–794, 2016. doi: 10.1145/2939672.2939785.

[27] K. Chivers. Zero-day vulnerability: What it is, and how it works. https://us.norton.com/internetsecurity-emerging-threats-how-do-zero-day-vulnerabilities-work-30sectech.html, August 2019. Accessed 2021-03-13.

[28] S. Christey and B. Martin. Buying into the bias: Why vulnerability statistics suck. https://txt.731my.com/tmp/blackHat/US-13-Martin-Buying-Into-The-Bias-Why-Vulnerability-Statistics-Suck-WP.pdf, 2013. Accessed 2021-02-21.

[29] P. Cichonski, D. Waltermire, and K. A. Kent. *Common Platform Enumeration: Dictionary Specification Version 2.3 (Draft)*. US Department of Commerce, National Institute of Standards and Technology, 2011.

[30] CISCO. What is cyber threat intelligence? https://www.cisco.com/c/en/us/products/security/what-is-cyber-threat-intelligence.html, 2021. Accessed 2021-03-21.

[31] S. Coble. CISA issues emergency vulnerability warning. https://www.infosecurity-magazine.com/news/cisa-issues-emergency/, July 2020. Accessed 2020-07-19.

[32] CollegeSimply. Virginia colleges ranked by largest enrollment. https://www.collegesimply.com/colleges/rank/colleges/largest-enrollment/state/virginia/, 2021. Accessed 2021-12-18.

[33] CVE-2019-0013 Routing Protocol Daemon (RPD) process will crash. National Vulnerability Database, January 2019. https://nvd.nist.gov/vuln/detail/CVE-2019-0013.

[34] CVE-2020-0240 Google Android RCE Flaw. National Vulnerability Database, August 2020. https://nvd.nist.gov/vuln/detail/CVE-2020-0240.

[35] CVE-2020-14334 Red Hat Satellite 6 Flaw. National Vulnerability Database, July 2020. https://nvd.nist.gov/vuln/detail/CVE-2020-14334.

[36] CVE-2020-9402 Django SQL Injection. National Vulnerability Database, August 2020. [online] https://nvd.nist.gov/vuln/detail/CVE-2020-9402.

[37] CVE-2021-23277 Eaton Intelligent Power Manager. National Vulnerability Database, April 2021. https://nvd.nist.gov/vuln/detail/CVE-2021-23277.

[38] CVE-2021-44228 Apache Log4j. National Vulnerability Database, August 2022. https://nvd.nist.gov/vuln/detail/CVE-2021-44228.

[39] Cybersecurity and Infrastructure Security Agency. Attack patterns. https://us-cert.cisa.gov/bsi/articles/knowledge/attack-patterns. Accessed 2021-02-20.

[40] Cybersecurity and Infrastructure Security Agency. Known exploited vulnerabilities catalog. https://www.cisa.gov/known-exploited-vulnerabilities-catalog. Accessed 2022-10-05.

[41] Cybersecurity and Infrastructure Security Agency. Critical infrastructure sectors. https://www.cisa.gov/critical-infrastructure-sectors, October 2020. Accessed 2022-10-08.

[42] Cybersecurity and Infrastructure Security Agency. Binding Operational Directive 22-01 Reducing the Significant Risk of Known Exploited Vulnerabilities. https://www.cisa.gov/binding-operational-directive-22-01, November 2021. Accessed 2022-10-05.

[43] Cybersecurity and Infrastructure Security Agency. CISA issues emergency directive requiring federal agencies to mitigate Apache Log4j vulnerabilities. https://www.cisa.gov/news/2021/12/17/cisa-issues-emergency-directive-requiring-federal-agencies-mitigate-apache-log4, December 2021. Accessed 2022-10-04.

[44] Cybersecurity and Infrastructure Security Agency. Review of the December 2021 Log4j event. https://www.cisa.gov/sites/default/files/publications/CSRB-Report-on-Log4-July-11-2022_508.pdf, July 2022. Accessed 2022-10-04.

[45] Cycode. CISA: the log4j vulnerability can impact hundreds of millions of devices... https://twitter.com/CycodeHQ/status/1476509636908003328, December 2021. Accessed 2022-10-04.

[46] D. Dey, A. Lahiri, and G. Zhang. Optimal policies for security patch management. *INFORMS Journal on Computing*, 27(3):462–477, 2015. doi: 10.1287/ijoc.2014.0638.

[47] Y. Dong, W. Guo, Y. Chen, X. Xing, Y. Zhang, and G. Wang. Towards the detection of inconsistencies in public security vulnerability reports. In *28th USENIX Security Symposium USENIX Security 19*, pages 869–885, 2019.

[48] T. Dumitraş and D. Shou. Toward a standard benchmark for computer security research: The Worldwide Intelligence Network Environment (WINE). In *Proceedings of the First Workshop on Building Analysis Datasets and Gathering Experience Returns for Security*, pages 89–96, 2011.

[49] A. Eitel. Environmental aware vulnerability scoring. In *International Conference on Internet of Things, Big Data and Security (IoTBDS)*, pages 478–485, 2020. doi: 10.5220/0009839104780485.

[50] F. Ekaputra. SEPSES Vocabulary. https://github.com/sepses/vocab. Accessed 2021-03-28.

[51] C. Elbaz, L. Rilling, and C. Morin. Fighting N-day vulnerabilities with automated CVSS vector prediction at disclosure. In *Proceedings of the 15th International Conference on Availability, Reliability and Security*, pages 1–10, 2020. doi: 10.1145/3407023.3407038.

[52] ELVIS. Security databases ecosystem — Embedded Lab Vienna for IoT and Security. https://wiki.elvis.science/index.php?title=Security_Databases_Ecosystem, 2020. Accessed 2020-02-14.

[53] K. Fazzini. In a decade of cybersecurity alarms, these are the breaches that actually mattered. https://www.cnbc.com/2019/12/23/stuxnet-target-equifax-worst-breaches-of-2010s.html, December 2019. Accessed 2022-10-25.

[54] W. M. Fitzgerald and S. N. Foley. Avoiding inconsistencies in the security content automation protocol. In *2013 IEEE Conference on Communications and Network Security (CNS)*, pages 454–461. IEEE, 2013.

[55] P. Foreman. *Vulnerability Management*. Taylor and Francis Group, 2010. ISBN: 978-1-4398-0150-5.

[56] Forum of Incident Response and Security Teams (FIRST). Common Vulnerability Scoring System version 3.1 specification document, revision 1. https://www.first.org/cvss/v3.1/specification-document, 2018. Accessed 2020-08-02.

[57] Forum of Incident Response and Security Teams (FIRST). Exploit Prediction Scoring System v2022.01.01. https://www.first.org/epss/, 2022. Accessed 2022-10-06.

[58] Forum of Incident Response and Security Teams (FIRST). Probability, percentiles, and binning - how to understand and interpret EPSS scores. https://www.first.org/epss/articles/prob_percentile_bins, 2022. Accessed 2022-10-06.

[59] S. Frei, M. May, U. Fiedler, and B. Plattner. Large-scale vulnerability analysis. In *Proceedings of the 2006 SIGCOMM workshop on Large-scale attack defense*, pages 131–138, 2006. doi: 0.1145/1162666.1162671.

[60] J. Freund and J. Jones. *Measuring and managing information risk: a FAIR approach*. Butterworth-Heinemann, 2014. ISBN: 9780124202313.

[61] C. Fruhwirth and T. Mannisto. Improving CVSS-based vulnerability prioritization and response with context information. In *2009 3rd International symposium on empirical software engineering and measurement*, pages 535–544. IEEE, 2009. doi: 10.1109/ESEM.2009.5314230.

[62] L. Gallon. On the impact of environmental metrics on CVSS scores. In *2010 IEEE Second International Conference on Social Computing*, pages 987–992. IEEE, 2010. doi: 10.1109/SocialCom.2010.146.

[63] L. Gallon. Vulnerability discrimination using CVSS framework. In *2011 4th IFIP International Conference on New Technologies, Mobility and Security*, pages 1–6. IEEE, 2011.

[64] L. Gallon and J.-J. Bascou. CVSS attack graphs. In *2011 Seventh International Conference on Signal Image Technology & Internet-Based Systems*, pages 24–31. IEEE, 2011. doi: 10.1109/SITIS.2011.24.

[65] L. Gallon and J. J. Bascou. Using CVSS in attack graphs. In *Proceedings of the 2011 Sixth International Conference on Availability, Reliability and Security*, pages 59–66. IEEE, 2011. doi: 10.1109/ARES.2011.18.

[66] E. Gamma, R. Helm, R. Johnson, and J. Vlissides. Design patterns: Abstraction and reuse of object-oriented design. In *European Conference on Object-Oriented Programming*, pages 406–431. Springer, 1993.

[67] GeeksForGeeks. What is Apache Log4j vulnerability? https://www.geeksforgeeks.org/what-is-apache-log4j-vulnerability/, February 2022. Accessed 2022-10-04.

[68] D. Geer and M. Roytman. Measuring vs. modeling. In *USENIX The Advanced Computing Systems Association*, volume 38, December 2013. Accessed 2020-08-01.

[69] V. GG. Exploit kits: Cybercriminals ultimate weapon. https://www.secpod.com/blog/exploit-kits-cybercriminals-brahmastra-a-nuclear-weapon/, January 2020. Accessed 2021-02-21.

[70] S. Ghosalkar. Basics of any critical infrastructure. https://etn-peter.eu/2021/04/06/basics-of-any-critical-infrastructure/, April 2021. Accessed 2022-10-08.

[71] Global Cybersecurity Alliance. Log4j vulnerability: What, why and how. https://gca.isa.org/blog/log4j-vulnerability-what-why-and-how. Accessed 2022-10-04.

[72] J. Goodall. STUCCO: Situation and Threat Understanding by Correlating Contextual Observations. https://stucco.github.io/. Accessed 2021-03-28.

[73] E. Hemberg, J. Kelly, M. Shlapentokh-Rothman, B. Reinstadler, K. Xu, N. Rutar, and U.-M. O'Reilly. Linking threat tactics, techniques, and patterns with defensive weaknesses, vulnerabilities and affected platform configurations for cyber hunting. *arXiv preprint arXiv:2010.00533*, 2020.

[74] J. Homer, X. Ou, and D. Schmidt. A sound and practical approach to quantifying security risk in enterprise networks. *Kansas State University Technical Report*, (2009-3), 2009. http://people.cs.ksu.edu/~xou/publications/tr_homer_0809.pdf.

[75] S. Horawalavithana, A. Bhattacharjee, R. Liu, N. Choudhury, L. O. Hall, and A. Iamnitchi. Mentions of security vulnerabilities on Reddit, Twitter and GitHub. In *IEEE/WIC/ACM International Conference on Web Intelligence*, pages 200–207, 2019.

[76] A. Horváth, P. M. Erdősi, and F. Kiss. The Common Vulnerability Scoring System (CVSS) generations–usefulness and deficiencies. In *Információs Társadalomért Alapítvány*, 2016. http://real.mtak.hu/93404/7/137-153%20attila_horvath_phd_E28093_peter_mate_erdosi_E28093_ferenc_kiss_phd_the_common_vulnerability_scoring_system_cvss_generations_E28093_usefulness_and_deficiencies.pdf.

[77] F. Howard. A closer look at the Angler exploit kit, July 2015. https://news.sophos.com/en-us/2015/07/21/a-closer-look-at-the-angler-exploit-kit//.

[78] M. Iannacone, S. Bohn, G. Nakamura, J. Gerth, K. Huffer, R. Bridges, E. Ferragut, and J. Goodall. Developing an ontology for cyber security knowledge graphs. In *Proceedings of the 10th Annual Cyber and Information Security Research Conference*, pages 1–4, 2015.

[79] Infoblox. An introduction to MITRE ATT&CK. https://blogs.infoblox.com/security/an-introduction-to-mitre-attck/, November 2019. Accessed 2021-02-21.

[80] Information Security Forum. Protecting the crown jewels: How to secure mission-critical assets. https://www.securityforum.org/solutions-and-insights/protecting-the-crown-jewels/, 2021. Accessed 2021-09-04.

[81] International Organization for Standardization. Information technology — security techniques — evaluation criteria for it security — part 1: Introduction and general model. https://www.iso.org/standard/50341.html, August 2022.

[82] International Standards Organization. ISO guide 73: 2009. *Risk management—Vocabulary*, 551:49, 2009. https://www.iso.org/standard/44651.html.

[83] J. Jacobs, S. Romanosky, I. Adjerid, and W. Baker. Improving vulnerability remediation through better exploit prediction. *2019 Workshop on the Economics of Information Security*, 2019. https://weis2019.econinfosec.org/wp-content/uploads/sites/6/2019/05/WEIS_2019_paper_53.pdf.

[84] J. Jacobs, S. Romanosky, I. Adjerid, and W. Baker. Improving vulnerability remediation through better exploit prediction. *Journal of Cybersecurity*, 6(1):tyaa015, 2020. doi: 10.1093/cybsec/tyaa015.

[85] J. Jacobs, S. Romanosky, B. Edwards, I. Adjerid, and M. Roytman. Exploit prediction scoring system (EPSS). *Digital Threats: Research and Practice*, 2(3):1–17, 2021.

[86] J. Jacobs, S. Romanosky, B. Edwards, M. Roytman, and I. Adjerid. Exploit prediction scoring system (EPSS). *arXiv preprint arXiv:1908.04856*, 2019.

[87] K. Järvelin and J. Kekäläinen. Cumulated gain-based evaluation of ir techniques. *ACM Transactions on Information Systems (TOIS)*, 20(4):422–446, 2002.

[88] P. Johnson, R. Lagerström, M. Ekstedt, and U. Franke. Can the common vulnerability scoring system be trusted? A Bayesian analysis. *IEEE Transactions on Dependable and Secure Computing*, 15(6):1002–1015, 2016. doi: 10.1109/TDSC.2016.2644614.

[89] C. L. Jones, R. A. Bridges, K. M. Huffer, and J. R. Goodall. Towards a relation extraction framework for cyber-security concepts. In *Proceedings of the 10th Annual Cyber and Information Security Research Conference*, pages 1–4, 2015. doi: 10.1145/2746266.2746277.

[90] A. Khazaei, M. Ghasemzadeh, and V. Derhami. An automatic method for CVSS score prediction using vulnerabilities description. *Journal of Intelligent & Fuzzy Systems*, 30(1):89–96, 2016.

[91] M. Khosravi-Farmad, R. Rezaee, and A. G. Bafghi. Considering temporal and environmental characteristics of vulnerabilities in network security risk assessment. In *2014 11th International ISC Conference on Information Security and Cryptology*, pages 186–191. IEEE, 2014. doi: 10.1109/ISCISC.2014.699404.

[92] E. Kiesling, A. Ekelhart, K. Kurniawan, and F. Ekaputra. The SEPSES knowledge graph: An integrated resource for cybersecurity. In *International Semantic Web Conference*, pages 198–214. Springer, 2019.

[93] J. Kirk. The Neutrino exploit kit has a new way to detect security researchers, Feb 2016. https://www.computerworld.com/article/3030418/the-neutrino-exploit-kit-has-a-new-way-to-detect-security-researchers.html.

[94] E. Kovacs. DHS orders agencies to patch critical vulnerabilities within 15 days. https://www.securityweek.com/dhs-orders-agencies-patch-critical-flaws-within-15-days, May 2019. Accessed 2020-07-16.

[95] C. C. Krebs. Emergency directive 20-03 (ED 20-03), mitigate windows DNS server vulnerability from July 2020 Patch Tuesday, July 2020. https://cyber.dhs.gov/ed/20-03/.

[96] R. P. Lippmann, K. W. Ingols, C. Scott, K. Piwowarski, K. Kratkiewicz, M. Artz, and R. Cunningham. Evaluating and strengthening enterprise network security using attack graphs. *Project Report IA-2, MIT Lincoln Laboratory*, 2005. https://apps.dtic.mil/dtic/tr/fulltext/u2/a439888.pdf.

[97] L. Lorimer. CPE Developer Day Workshop. https://www.slideserve.com/lynnea/developer-web-conference-powerpoint-ppt-presentation, May 2010. Accessed 2021-03-20.

[98] M. Mago and F. F. Madyira. Ransomware software: Case of WannaCry. *Engineering and Science*, 3(1):258–261, 2018.

[99] C. D. Manning. *Introduction to information retrieval*. Cambridge University Press, 2008.

[100] R. A. Martin. Common Weakness Enumeration. https://cwe.mitre.org/, 2007. Accessed 2021-01-15.

[101] V. Mavroeidis, R. Hohimer, T. Casey, and A. Jesang. Threat actor type inference and characterization within cyber threat intelligence. In *2021 13th International Conference on Cyber Conflict (CyCon)*, pages 327–352. IEEE, 2021.

[102] L. McBride. Major government attack highlights how Log4j is still unresolved. https://blog.sonatype.com/major-government-attack-highlights-log4j-resolution-shortfall, March 2022. Accessed 2022-10-04.

[103] P. Mell, K. Scarfone, and S. Romanosky. A complete guide to the Common Vulnerability Scoring System version 2.0. In *FIRST-forum of incident response and security teams*, volume 1, pages 1–23, 2007.

[104] Microsoft Security Response Center. Microsoft Security Update Guide. `https://msrc.microsoft.com/update-guide/vulnerability`, January 2021. Accessed 2021-01-24.

[105] J. D. Mireles, J.-H. Cho, and S. Xu. Extracting attack narratives from traffic datasets. In *Proceedings of the 2016 International Conference on Cyber Conflict (CyCon US)*, pages 1–6. IEEE, 2016.

[106] MITRE ATT&CK Enterprise, Version 10.1. `https://web.archive.org/web/20220215084226/https://attack.mitre.org/`, 2022. Accessed 2022-02-17.

[107] MITRE ATT&CK Enterprise, Version 9. `https://web.archive.org/web/20210501085039/https://attack.mitre.org/`, 2021. Accessed 2021-05-01.

[108] MITRE Corporation. Common Vulnerabilities and Exposures. `https://cve.mitre.org/`, 2021.

[109] S. Narayanan, A. Ganesan, K. Joshi, T. Oates, A. Joshi, and T. Finin. Cognitive techniques for early detection of cybersecurity events. *arXiv preprint arXiv:1808.00116*, 2018.

[110] K. Nayak, D. Marino, P. Efstathopoulos, and T. Dumitraş. Some vulnerabilities are different than others. In *International Workshop on Recent Advances in Intrusion Detection*, pages 426–446. Springer, 2014.

[111] Neo4j. What is a graph database? `https://neo4j.com/developer/graph-database/`. Accessed 2021-06-05.

[112] V. H. Nguyen and F. Massacci. The (un) reliability of NVD vulnerable versions data: An empirical experiment on Google Chrome vulnerabilities. In *Proceedings of the 8th ACM SIGSAC symposium on Information, computer and communications security*, pages 493–498, 2013.

[113] NIST. CVSS Severity over time. `https://nvd.nist.gov/general/visualizations/vulnerability-visualizations/cvss-severity-distribution-over-time`. Accessed 2021-02-07.

[114] NIST. National Vulnerability Database. https://nvd.nist.gov/vuln-metrics/cvss. Accessed 2020-12-05.

[115] NIST. NVD Dashboard. https://nvd.nist.gov/general/nvd-dashboard. Accessed 2021-02-13.

[116] NIST. Vulnerability type change by year. https://nvd.nist.gov/general/visualizations/vulnerability-visualizations/cwe-over-time#vuln-type-change-by-year-desc. Accessed 2021-02-07.

[117] S. Noel, E. Harley, K. H. Tam, and G. Gyor. Big-data architecture for cyber attack graphs. *MITRE case number 14-3549*, 2014. https://csis.gmu.edu/noel/pubs/2015_IEEE_HST.pdf.

[118] S. Noel, E. Harley, K. H. Tam, and G. Gyor. Big-data architecture for cyber attack graphs representing security relationships in NoSQL graph databases. In *IEEE Symposium on Technologies for Homeland Security (HST)*, April 2015.

[119] G. R. Notess. The Wayback Machine: The web's archive. *Online*, 26(2):59–61, 2002.

[120] B. Obama. Presidential Policy Directive 21 (PPD-21) on critical infrastructure security and resilience, February 2013. https://infragardsd.org/docs/ppd21.pdf.

[121] J. L. Obes, C. Sarraute, and G. Richarte. Attack planning in the real world. *arXiv preprint arXiv:1306.4044*, 2013.

[122] L. O'Donnell. High-severity Android RCE flaw fixed in August security update. https://threatpost.com/high-severity-android-rce-flaw-fixed-in-august-security-update/158049/, August 2020. Accessed 2020-12-19.

[123] L. O'Donnell. Microsoft pulls bad windows update after Patch Tuesday headaches. https://threatpost.com/microsoft-windows-update-patch-tuesday/163981/, February 2021. Accessed 2021-03-13.

[124] X. Ou, W. F. Boyer, and M. A. McQueen. A scalable approach to attack graph generation. In *Proceedings of the 13th ACM Conference on Computer and Communications Security*, pages 336–345, 2006.

[125] X. Ou, S. Govindavajhala, and A. W. Appel. MulVal: A logic-based network security analyzer. In *USENIX security symposium*, volume 8, pages 113–128. Baltimore, MD, 2005.

[126] S. Ozkan. CVE details: The ultimate cybersecurity vulnerability datasource. `https://www.cvedetails.com`, December 2020. Accessed 2020-12-20.

[127] PCI, Payment Card Industry. Data security standard. *Requirements and Security Assessment version*, 3, 2010.

[128] O. I. Poyraz, S. Bouazzaoui, O. Keskin, M. McShane, and C. A. Pinto. Cyber-assets at risk (CAR): The cost of personally identifiable information data breaches. In *Proceedings of the International Conference on Cyber Warfare and Security*, pages 402–XVI. Academic Conferences International Limited, 2020.

[129] S. Quinn, K. Scarfone, and D. Waltermire. Guide to adopting and using the Security Content Automation Protocol (SCAP) version 1.2. Technical Report NIST SP 800-117, National Institute of Standards and Technology, 2012. `https://tsapps.nist.gov/publication/get_pdf.cfm?pub_id=905179`.

[130] C. Research. APT35 exploits Log4j vulnerability to distribute new modular PowerShell toolkit. `https://txt.731my.com/tmp/blackHat/US-13-Martin-Buying-Into-The-Bias-Why-Vulnerability-Statistics-Suck-WP.pdf`, January 2022. Accessed 2022-10-04.

[131] I. Robinson, J. Webber, and E. Eifrem. *Graph databases: new opportunities for connected data*. O'Reilly Media, Inc., 2015. ISBN: 9781491930892.

[132] G. Roldán-Molina, M. Almache-Cueva, C. Silva-Rabadão, I. Yevseyeva, and V. Basto-Fernandes. A comparison of cybersecurity risk analysis tools. *Procedia computer science*, 121:568–575, 2017.

[133] D. M. Ross, A. B. Wollaber, and P. C. Trepagnier. Latent feature vulnerability ranking of CVSS vectors. In *Proceedings of the Summer Simulation Multi-Conference*, pages 1–12, 2017. doi: `10.5555/3140065.3140084`.

[134] C. Sabottke, O. Suciu, and T. Dumitraș. Vulnerability disclosure in the age of social media: Exploiting Twitter for predicting real-world exploits. In *24th USENIX Security Symposium Security 15*, pages 1041–1056, 2015.

[135] T. Sager. The cyber OODA loop: How your attacker should help you design your defense. https://csrc.nist.gov/CSRC/media/Presentations/The-Cyber-OODA-Loop-How-Your-Attacker-Should-Help/images-media/day3_security-automation_930-1020.pdf, 2015. Accessed 2020-08-01.

[136] L. A. B. Sanguino and R. Uetz. Software vulnerability analysis using CPE and CVE. *arXiv preprint arXiv:1705.05347*, 2017.

[137] SANS Offensive Operations. Microsoft just released a patch for critical vuln SIGRed... https://twitter.com/SANSOffensive/status/1283389702410772483, July 2020. Accessed 2020-11-20.

[138] O. Santos. The evolution of scoring security vulnerabilities: The sequel. https://blogs.cisco.com/security/cvssv3-study, October 2016. Accessed 2020-12-13.

[139] K. Scarfone and P. Mell. An analysis of CVSS version 2 vulnerability scoring. In *Proceedings of the 2009 3rd International Symposium on Empirical Software Engineering and Measurement*, pages 516–525. IEEE, 2009.

[140] M. Scheck. Example of a CVSS based patch policy for an enterprise. https://www.first.org/cvss/v2/cvss-based-patch-policy.pdf, 2021. Accessed 2021-02-14.

[141] N. Schmeidler. Microsoft Patch Tuesday: All or nothing patches. https://blog.morphisec.com/microsoft-patch-tuesday-all-or-nothing-patching, October 2016. Accessed 2022-10-25.

[142] B. Schneier. Should U.S. hackers fix cybersecurity holes or exploit them? https://www.theatlantic.com/technology/archive/2014/05/should-hackers-fix-cybersecurity-holes-or-exploit-them/371197, May 2014. Accessed 2020-07-30.

[143] A. Scroxton. China's APT41 exploited Log4j within hours. https://www.computerweekly.com/news/252514376/Chinas-APT41-exploited-Log4j-within-hours, March 2022. Accessed 2022-10-04.

[144] O. Sheyner, J. Haines, S. Jha, R. Lippmann, and J. M. Wing. Automated generation and analysis of attack graphs. In *Proceedings of the 2002 IEEE Symposium on Security and Privacy*, pages 273–284. IEEE, 2002. doi: 10.1109/SECPRI.2002.1004377.

[145] R. Simmons. BYOD Security Implementation for Small Organizations. White Paper, SANS Institute Reading Room, Dec 2017. https://www.sans.org/reading-room/whitepapers/mobile/byod-security-implementation-small-organizations-38230/.

[146] H. A. Simon. Designing organizations for an information-rich world. In *Computers, communications, and the public interest*, pages 37–72, 1969. http://opacplus.bsb-muenchen.de/search?isbn=0-8018-1135-X.

[147] A. Singhal and X. Ou. Security risk analysis of enterprise networks using probabilistic attack graphs. In *Network Security Metrics*, pages 53–73. Springer, 2017. doi: 10.1007/978-3-319-66505-4_3.

[148] B. E. Strom, A. Applebaum, D. P. Miller, K. C. Nickels, A. G. Pennington, and C. B. Thomas. MITRE ATT&CK: Design and philosophy. *Technical report*, (MP180360R1), 2018. https://www.mitre.org/sites/default/files/publications/pr-18-0944-11-mitre-attack-design-and-philosophy.pdf.

[149] Y. Tale. CVE-2020-14334 foreman: unauthorized cache read on RPM-based installations through local user. https://bugzilla.redhat.com/show_bug.cgi?id=1858284, July 2020. Accessed 2020-12-19.

[150] J. Tang, S. Alelyani, and H. Liu. Feature selection for classification: A review. *Data Classification: Algorithms and Applications*, pages 37–64, 2014.

[151] M. Tatam, B. Shanmugam, S. Azam, and K. Kannoorpatti. A review of threat modelling approaches for APT-style attacks. *Heliyon*, 7(1):e05969, 2021.

[152] Y. Tatarinova and O. Sinelnikova. Extended vulnerability feature extraction based on public resources. *Theoretical and applied cybersecurity*, 1(1), 2019.

[153] TheHackerNews. A critical 17-year-old 'wormable' RCE vulnerability affects Windows DNS servers... https://twitter.com/TheHackersNews/status/1283087444015935489, July 2020. Accessed 2020-11-20.

[154] TheHackerNews. Mcrosoft warns of continued attempts by nation-state adversaries... https://twitter.com/TheHackersNews/status/1478629860041887744, January 2022. Accessed 2022-10-04.

[155] Trend Micro. After WannaCry, UIWIX ransomware follows suit. https://www.trendmicro.com/en_us/research/17/e/wannacry-uiwix-ransomware-monero-mining-malware-follow-suit.html, May 2017. Accessed 2021-01-24.

[156] R. Tsang. 100+ vulnerabilities patched during Patch Tuesdays the new norm. https://blog.rapid7.com/2020/07/15/patch-tuesday-july-2020/, July 2020. Accessed 2020-07-14.

[157] E. Tsukerman. Cybersecurity Threat Modeling with OCTAVE. https://www.pluralsight.com/guides/cybersecurity-threat-modeling-with-octave, September 2020. Accessed 2021-05-20.

[158] U.S. Department of State. Independent states in the world. https://www.state.gov/independent-states-in-the-world/, September 2022. Accessed 2022-10-08.

[159] US Election Assistance Commission. CI Scoop: What are sectors and sub-sectors? https://www.eac.gov/ci-scoop-what-are-sectors-and-sub-sectors, May 2017. Accessed 2022-10-08.

[160] M. Van Horenbeeck. The SANS Internet Storm Center. In *Proceedings of the 2008 WOMBAT Workshop on Information Security Threats Data Collection and Sharing*, pages 17–23. IEEE, 2008.

[161] Vuldb. Eaton intelligent power manager up to 1.68 code syntax scripts/libs/utils.js loaduserfile code injection. https://vuldb.com/?id.172986, April 2021. Accessed 2021-05-16.

[162] D. Waltermire, S. Quinn, H. Booth, K. Scarfone, and D. Prisaca. The Technical Specification for the Security Content Automation Protocol (SCAP): SCAP Version 1.3. Technical Report NIST SP 800-126, National Institute of Standards and Technology, 2016.

[163] J. A. Wang and M. Guo. OVM: An ontology for vulnerability management. In *Proceedings of the 5th Annual Workshop on Cyber Security and Information Intelligence Research: Cyber Security and Information Intelligence Challenges and Strategies*, pages 1–4, 2009. doi: 10.1145/1558607.1558646.

[164] L. Wang, T. Islam, T. Long, A. Singhal, and S. Jajodia. An attack graph-based probabilistic security metric. In *Proceedings of the IFIP Annual Conference on Data*

and Applications Security and Privacy, pages 283–296. Springer, 2008. doi: 10.1007/978-3-540-70567-3_22.

[165] Wikipedia. Open Source Vulnerability Database (OSVDB). https://en.wikipedia.org/wiki/Open_Source_Vulnerability_Database. Accessed 2020-05-01.

[166] Wikipedia. Patch Tuesday — Wikipedia, the free encyclopedia. http://en.wikipedia.org/w/index.php?title=Patch%20Tuesday&oldid=992184086, 2020. Accessed 2020-12-05.

[167] Wikipedia. Petya (malware) — Wikipedia, the free encyclopedia. http://en.wikipedia.org/w/index.php?title=Petya%20(malware)&oldid=984457286, 2020. Accessed 2020-12-05.

[168] Wikipedia. WannaCry ransomware attack — Wikipedia, the free encyclopedia. http://en.wikipedia.org/w/index.php?title=WannaCry%20ransomware%20attack&oldid=991892624, 2020. Accessed 2020-12-05.

[169] Wikipedia. Bulletproof hosting. https://en.wikipedia.org/wiki/Bulletproof_hosting, 2021. Accessed 2021-01-24.

[170] Wikipedia. Fancy Bear — Wikipedia, the free encyclopedia. https://en.wikipedia.org/wiki/Fancy_Bear, 2021. Accessed 2021-06-12.

[171] Wikipedia. Internet Kill Switch — Wikipedia, the free encyclopedia. https://en.wikipedia.org/wiki/Internet_kill_switch, 2021. Accessed 2021-01-24.

[172] Wikipedia. Mirai (malware) — Wikipedia, the free encyclopedia. https://en.wikipedia.org/wiki/Mirai_(malware), 2021. Accessed 2021-01-24.

[173] Wikipedia. Student's T-Test. https://en.wikipedia.org/wiki/Student%27s_t-test, 2022. Accessed 2022-11-17.

[174] K. H. Zou, A. J. O'Malley, and L. Mauri. Receiver-operating characteristic analysis for evaluating diagnostic tests and predictive models. *Circulation*, 115(5):654–657, 2007. doi: 10.1161/CIRCULATIONAHA.105.594929.

APPENDIX A

EDUCATION SUBSECTOR SOFTWARE LIST

Table 48: Full education subsector software list for Virginia Universities. Product names are listed exactly as they appeared on the university's website.

W&M	ODU	VT	RU	UVA	WLU
Adobe Acrobat Reader DC	7-Zip	Adobe Creative Cloud	Adobe Acrobat Reader Dc	Adobe Fonts	After Effects
Advance	AGI/ STK11	Autodesk	Adobe Acrobat Pro Dc	Adobe Reader	Illustrator
Alertus Desktop Notification	Adobe Acrobat Professional	Duo D-100 Token	Adobe Creative Cloud	Alertus	Photoshop
AutoDesk	Adobe After Effects CC	ESRI ArcGIS Desktop Education Edition	Adobe Spark	ANSYS	Premiere
Banner Admin Production	Adobe Animate	EquatIO	Application Xtender	ArcGIS Suite	Audacity
Box	Adobe Creative Cloud	GRAHL PDF Annotator	Autodesk Maya	Dedoose	Autodesk Inventor
Box Drive	Adobe Dream weaver CC	Granta CES Edupack	Citrix	Eventbrite	Arc GIS
Chem BioDraw Ultra	Adobe Illustrator CC	MathWorks MATLAB	Final Draft	Firefox	Canvas

Table 48 Continued

W&M	ODU	VT	RU	UVA	WLU
Cylance Protect	Adobe InDesign CC	Minitab	Google Chrome	IDL	Skype
DUO Mobile	Adobe Lightroom CC	National Instruments LabVIEW	Google File Stream	LabView	Zoom
Eduroam	Adobe Photoshop CC	Qualitrcs (web-base)	Grammarly	LastPass	Cinema 4D
Global Protect for MacOs	Adobe Premiere Pro CC	QuestionPro (web-based)	Ibm SPSS	Wolfram Mathematica	Digication
Global Protect (VPN) for Windows	Adobe Reader	Read&Write	Junos Pulse (Vpn)	MATLAB	Final Cut Pro
Google Chrome	ArcGIS	Rhino	Microsoft Office	MetaAccess	FlipGrid
LinkedIn Learning	Arena	SAS	Microsoft Project	Microsoft Defender for Endpoint for Mac	iMovie
Maple	Audacity	SAS JMP Pro	Microsoft Teams	Microsoft Desktop Optimization Pack (MDOP)	Linkedin Learning
Matlab	Autodesk AutoCAD	SimaPro	Microsoft Visio	Microsoft Office	MeshLab
Microsoft Office 365	Autodesk Inventor	Solidworks	Minitab	Microsoft OneNote	Meta Shape
Microsoft Teams	Crestron AirMedia	SPSS	Movie Magic	Microsoft Project	Microsoft Forms

Table 48 Continued

W&M	ODU	VT	RU	UVA	WLU
Minitab for Windows	ERDAS Imagine	Windows Virtualization Add-on License	Mozilla Firefox	Microsoft Software & Tools for Learning	Wordpress
Minitab for Mac	EndNote	Wolfram Mathematica	Panopto	Microsoft System Center Endpoint Protection	Yuja
Mitel Connect for Mac	FastX		Power BI Pro	Microsoft Visio	Poll Everywhere
Mitel Connect for Windows	FileZilla		Respondus	Microsoft Windows	
Mozilla Firefox Mac	Geometers Sketchpad		Respondus Lockdown Browser	Minitab	
Mozilla Firefox Windows	Global Protect VPN Client		Snagit	Network Setup Tool	
Panopto	Google Calendar		Techsmith Camtasia	Oracle Java	
PCModel	Google Chrome		Toon Boom	OriginPro	
SAS	Google Drive		VLC Media Player	oXygenXML	
SPSS	Google Hangouts		Wrike	Perceptive Content	
Stata	Java		Zoom	Personal Digital Certificate	

Table 48 Continued

W&M	ODU	VT	RU	UVA	WLU
Wolfram Mathematica	LabVIEW			Qualtrics Research Suite	
Zoom	LiveSafe			Read&Write	
	MATLAB			SAS	
	MSC Patran/Nastran			Secure Deletion Shredder	
	MathCAD			SecureCRT	
	Microsoft Office			SecureFX	
	Microsoft Project			Smartsheet	
	Microsoft SQL Server Standard			SolidWorks	
	Microsoft SharePoint			SPSS	
	Microsoft Silverlight			SPSS AMOS	
	Microsoft Visio			Stata	
	Microsoft Windows Server Standard			StatTransfer	
	Minitab			UVA Anywhere	
	Mozilla Firefox			UVA Certificates Bundle	

Table 48 Continued

W&M	ODU	VT	RU	UVA	WLU
	NI Multisim			Visual Studio	
	NVivo			VMWare View (Hive)	
	Pearson POM/QM			VPN Client	
	RISA			Zoom	
	Remote Desktop				
	Respondus				
	Respondus Lockdown Browser				
	SAS Annual License 9.4 M6				
	SPSS Statistics				
	SSI HLM 7				
	Systat				
	Systat SigmaPlot				
	TurningPoint Student Response System				
	VLC Media Player				
	Web Drive				
	Web Xtender				
	WinAuth				

Table 48 Continued

W&M	ODU	VT	RU	UVA	WLU
	WinSCP				
	Windows 10				
	Wolfram Mathemat-ica				
	X-Win32				
	Zoom				

APPENDIX B

GOVERNMENT FACILITIES SOFTWARE LIST

Table 49: Full generated software list for government facilities.

GOV-S	GOV-M	GOV-L	GOV-XL
aruba 2930f	application delivery controller	altalink c8070 firmware	adaptive security appliance
cloud access manager	aruba 2930f	anyconnect secure mobility client	advanced threat defense
core	carbon black app control	catalyst ie3200 rugged switch	altalink c8070 firmware
extremexos	cloud access manager	clearpass	anyconnect secure mobility client
gigavue	core	cloud access manager	avocent umg-4000 firmware
globalprotect	cx 6200f	core	carbon black app control
knox	cx 8320	data protector	catalyst ie3200 rugged switch
m-200	cyber backup	desktop password reset	clearpass
m-500	deception	display solutions	cloud access manager
securitycenter	display solutions	dm-nvx-dir-160	cloud services router 1000v
unified endpoint management	enterprise linux eus	enterprise linux eus	clustered data ontap
wf-500	extremexos	esr6300	core
workspace one	fireware xtm	extremexos	cx 6200f
	galaxy apps	galaxy apps	cx 8320
	gigavue	gigavue	data protector

Table 49 Continued

GOV-S	GOV-M	GOV-L	GOV-XL
	global vpn client	global vpn client	deception
	m-200	globalprotect	desktop password reset
	nessus	m-100	display solutions
	nessus network monitor	m-200	dm-nvx-dir-160
	next generation firewall	mac os x	enterprise linux eus
	purity	nessus agent	esr6300
	secure mobile access	nessus network monitor	extremexos
	stealthwatch enterprise	next generation firewall	fortiwlc
	wf-500	pa-220	galaxy apps
		pixel	gigavue
		securitycenter	global vpn client
		splunk	globalprotect
		stealthwatch enterprise	intrusion prevention system
		unified endpoint management	m-100
		workspace one	m-200
			m-500
			mac os x
			ncs 1001
			nessus
			nessus agent
			nessus network monitor
			next generation firewall
			pa-220

Table 49 Continued

GOV-S	GOV-M	GOV-L	GOV-XL
			pdumh15at
			pixel
			security gateway
			securitycenter
			splunk
			stealthwatch enterprise
			unified endpoint management
			wf-500
			workspace one

VITA

Corren G. McCoy
Department of Computer Science
Old Dominion University
Norfolk, VA 23529

EDUCATION

Doctor of Philosophy, Computer Science (2022)
Old Dominion University, Norfolk, Virginia
Dissertation: *A Relevance Model for Threat-Centric Ranking of Cybersecurity Vulnerabilities*

Master of Arts, Management (2006)
Regent University, Virginia Beach, VA

Master of Science, Computer Science (1990)
Old Dominion University, Norfolk, Virginia
Project: *Multilevel Secure Databases: A New Approach*

Bachelor of Science, Computer Science (1983)
Pennsylvania State University, University Park, PA

PUBLICATIONS

An updated list of publications is available on Google Scholar at `https://scholar.google.com/citations?user=gp6cdH8AAAAJ`

Typeset using LaTeX.