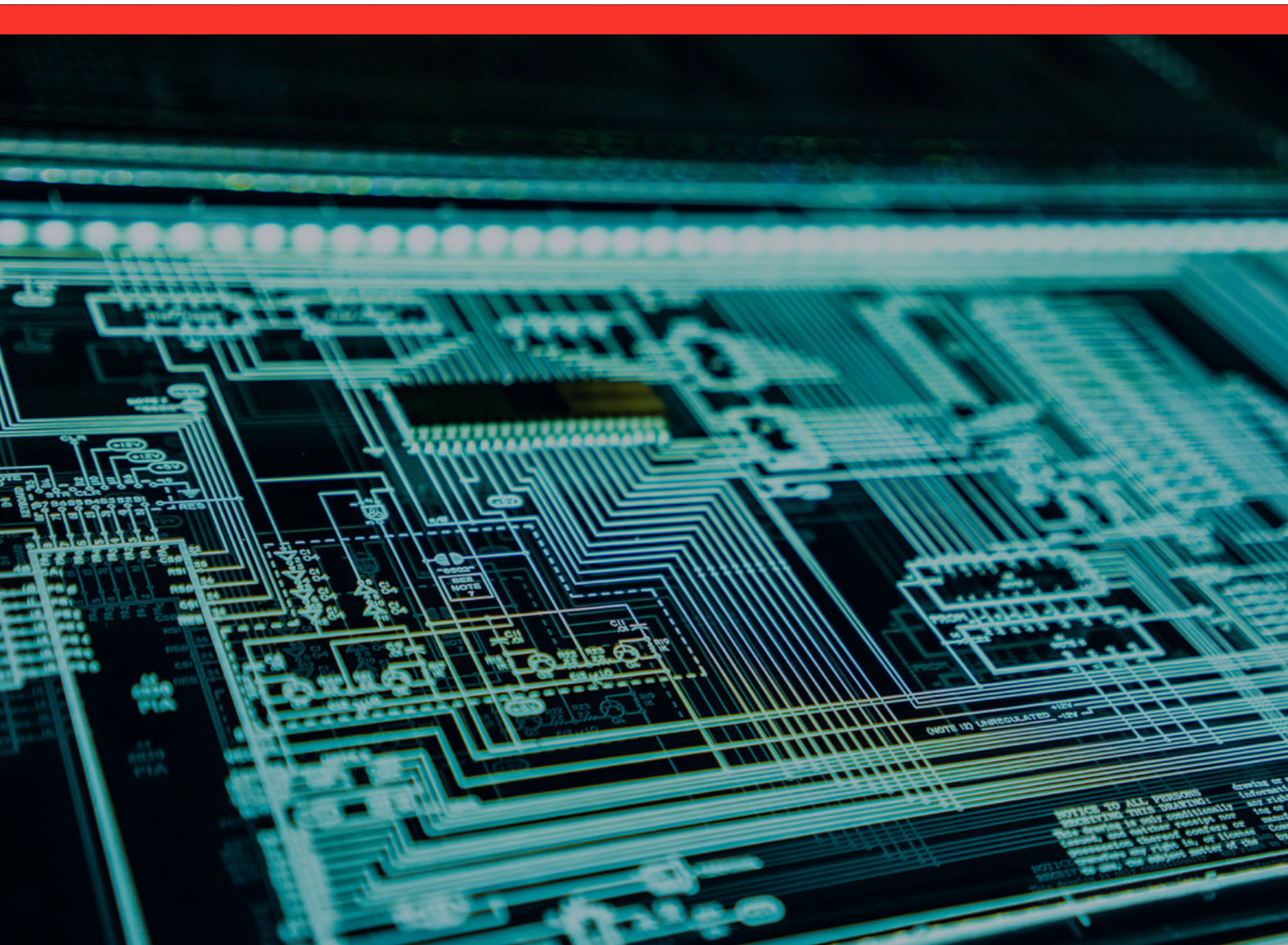
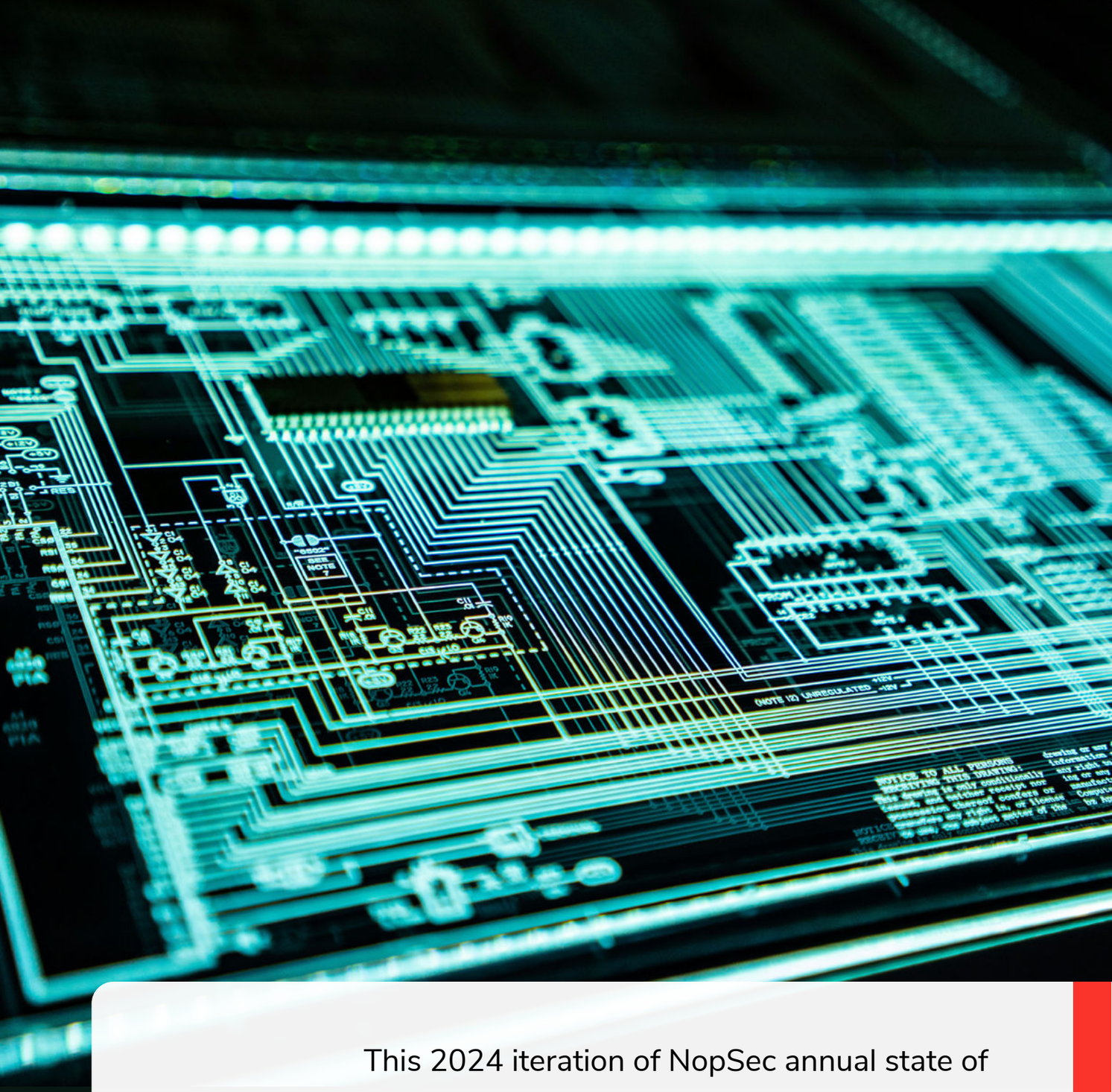
NOPSEC

2024

State of Threat and Exposure Management Report





This 2024 iteration of NopSec annual state of vulnerability management focuses on the crossroad between threat-based prioritization and contextual risk management, delving deeper on how NopSec Risk Score prioritizes riskier vulnerabilities and how vulnerabilities CVE and ATT&CK framework can be mapped together to expand the threat and contextual risk vision of attack surface management. This report is divided into three parts: how the vulnerability landscape is measured in terms of vulnerability prevalence and remediation efficiency, how the NopSec Risk Score works in real-life vulnerability scenarios, and how CVE and ATT&CK taxonomies are mapped to expand the risk scenarios beyond vulnerability management.

2024 State of Threat and Exposure Management Report

TABLE OF CONTENTS

VOLUME 1

VOLUME 2

TABLE OF CONTENTS

VOLUME 3

APPENDIX

2024 State of Threat and Exposure Management Report

Analysis for Section 1 of this report was provided by the Cyentia Institute. Cyentia is a research and data science firm working to advance cybersecurity knowledge and practice. We do this by partnering with security vendors and other organizations to publish a range of high-quality, data-driven content like this study. Find out more at www.cyentia.com

VOLUME 1:

MEASURING THE VULNERABILITY LANDSCAPE

An independent analysis of real-world vulnerability and remediation data in collaboration with the Cyentia Institute

THIS SECTION analyzes 36 million vulnerabilities detected by organizations using NopSec to help prioritize remediation efforts. We begin by examining the prevalence of those vulnerabilities across assets to determine which ones are most common. Then we measure how quickly those vulnerabilities are remediated and what factors speed up or slow down that process. The section closes by identifying examples of vulnerabilities that are prone to slip through the cracks of traditional prioritization strategies so your organization won't waste valuable time and energy chasing after the wrong things.

VULNERABILITY PREVALENCE

We'll begin our foray into the wilds of the vulnerability landscape by examining the product vendors that shape it. This is important because these technologies are commonly used, thus vulnerabilities affecting them can have a widespread impact on cyber risk posture.

We measure vulnerability prevalence in three basic ways:

01. **Vulnerabilities published on the CVE List,**

02. **Enterprise assets affected by those vulnerabilities, and**

03. **The total number of detected vulnerabilities (open and closed) across all assets.**

Each of these measures offers a different perspective of the vulnerability landscape and is captured in the columns of Figure 1.

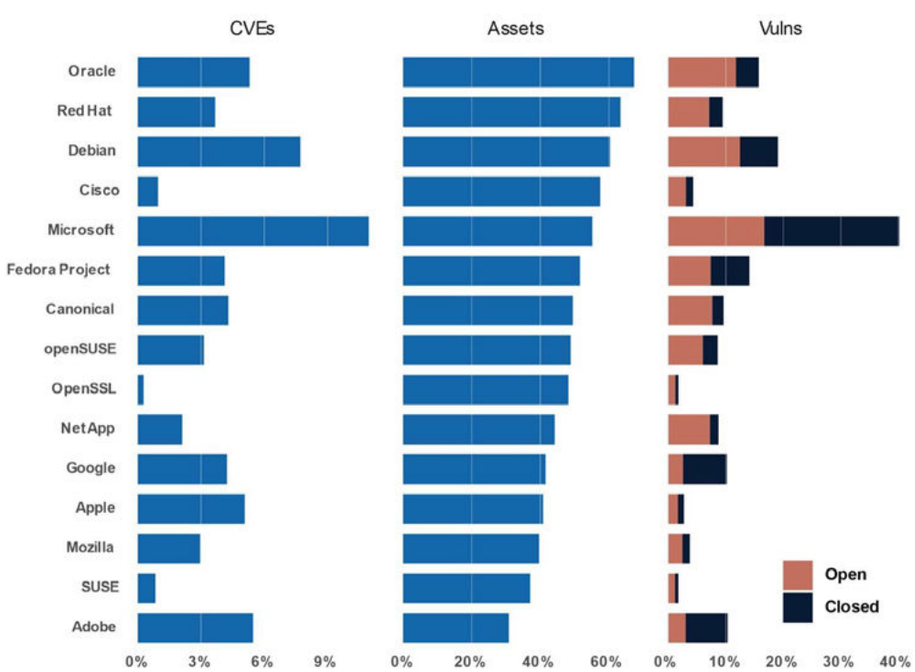


Figure 1: Prevalence of 15 most common vendors across CVEs, assets, and vulnerabilities

Microsoft immediately jumps out from Figure 1. That single vendor is behind 11% of all published CVEs, which together affect 55% of all organizational assets and comprise 40% of all vulnerabilities detected. Debian, Oracle, RedHat, Adobe, and Apple are also worth highlighting since those vendors make the top five in at least one of the columns in Figure 4.

We need to be careful, however, about jumping to the conclusion that many vulnerabilities equate to insecure or poorly managed products. These are big numbers, to be sure, but not too surprising given their massive footprint. Plus, numerous vulnerabilities can actually be a sign of vendors making a concerted effort to identify, disclose, and fix issues affecting their products.

Speaking of fixing issues, the rightmost column in the chart shows the ratio of open and closed vulnerabilities. This makes it easy to discern that some vendors (e.g., Google and Adobe) boast high closure rates, whereas others show the opposite (e.g., Oracle and NetApp). Overall, 49% of all vulnerabilities had been closed by the time we pulled data for this analysis. We'll dive deeper into remediation rates in a moment.

VULNERABILITY EXPLOITATION

When a new vulnerability emerges, questions like "Is it being exploited?" and "How exploitable is it?" quickly follow. Given that, we figured a view of various vulnerability factors that help answer such questions would be appropriate for this analysis. We list several in Figure 2.

Figure 2: Commonality of vulnerability factors relevant to exploitation

There's a lot to soak in from Figure 2. In the top half, we see that vulnerabilities that enable phishing and/or remote attacks are quite common, affecting over half of all organizational assets. The enablement of lateral movement is (thankfully) less common among CVEs and detected vulnerabilities but still affects a quarter of assets.

The last trait listed in Figure 2. concerns the celebrity status of vulnerabilities. It's quite rare among the plethora of published CVEs and even detected vulnerabilities. But over 20% of assets are plagued with a celebrity vuln. Like Hollywood celebrities, they tend to capture a lot of media attention. But are they the ones you should be paying attention to? More to come on that.

We see the same basic pattern for the existence of functional exploits and malware—they are less common among all vulnerabilities but very common across assets. And that's arguably what matters most since protecting assets is the goal of remediating vulnerabilities.

RISK RATINGS

With so many open vulnerabilities placing organizational assets at risk, prioritizing remediation efforts has surged to the forefront of VM strategy. That's usually done by identifying which vulnerabilities represent the highest risk, but how to do that best isn't always clear. Thus, many organizations default to CVSS for determining which vulnerabilities warrant priority remediation.

Figure 3 shows the downside of that strategy. According to CVSS, over 70% of all detected vulnerabilities are rated high or critical severity. That's unrealistic from a risk assessment standpoint and sets unrealistic expectations for remediation as well. Few organizations have the capacity to handle so many vulnerabilities.

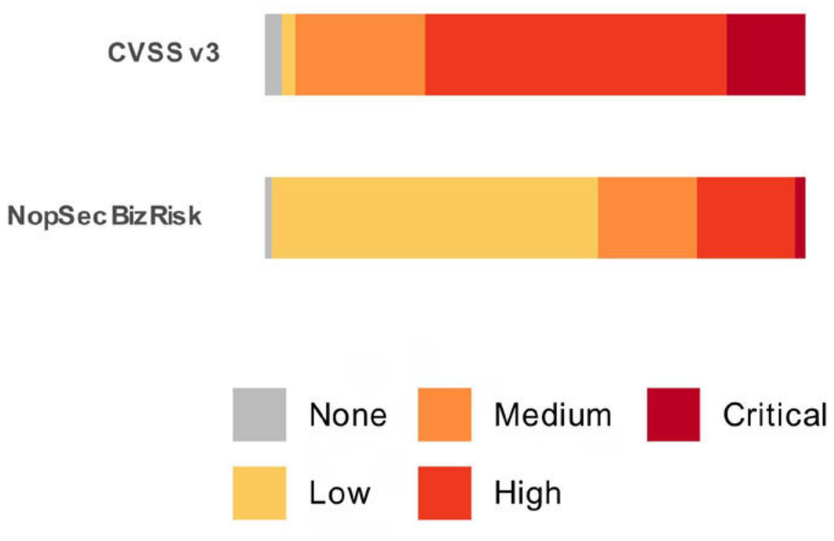


Figure 3: Breakdown of CVSSv3 and NopSec Business Risk Scores for detected vulnerabilities

That's why NopSec assigns a Business Risk Score to each vulnerability so customers can better prioritize their remediation efforts. The lower chart in Figure 3 shows a breakdown of detected vulnerabilities according to this score. A much more manageable 20% of vulnerabilities are rated as high or critical. In a later section, we'll demonstrate how this contributes to significantly improved remediation rates over CVSS.

REMEDIATION TIMELINES

We mentioned in the last section that 49% of all detected vulnerabilities had been successfully remediated. But how long did that process take and how does it vary across different types of vulnerabilities? That's exactly what we explore in this section.

> *There are multiple ways to measure how quickly vulnerabilities are closed, with mean-time-to-remediation (MTTR) using a straight average being the most common. There are several issues with that approach, however, which is why we favor a technique called survival analysis. It's more appropriate for long-tailed distributions like remediation timelines and handles the fact that so many vulnerabilities haven't yet been closed ("censored data").*

Based on a survival analysis of all vulnerabilities in our dataset, we calculate the overall average time-to-remediation at 212 days. This dying off (closing out) of vulnerabilities over time is traced by the dark gray "Overall" curve in Figure 4. But as you probably suspect, that timeline varies dramatically across organizations, assets, and different types of vulnerabilities.

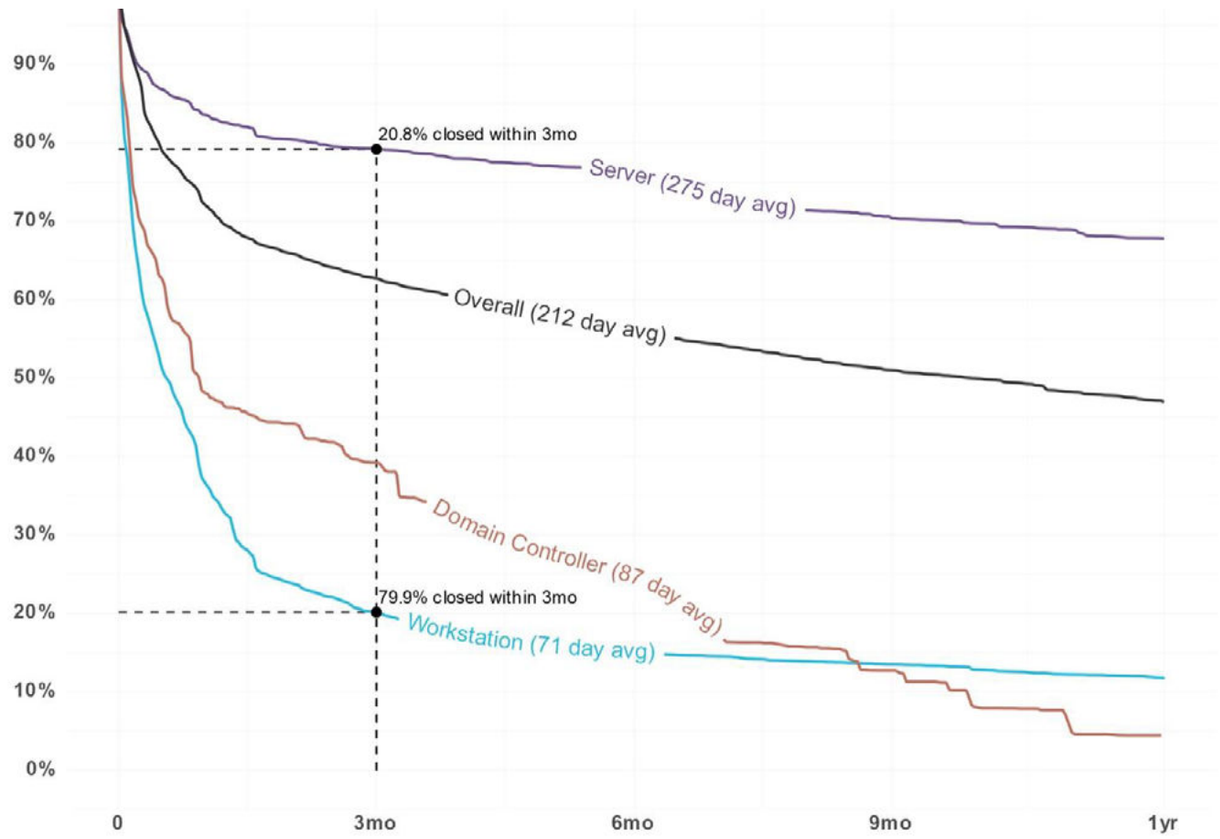


Figure 4: Comparison of vulnerability remediation timelines among asset categories

Figure 4 offers an example of such variation, comparing remediation timelines for three different categories of assets. Here we see that 80% of vulnerabilities affecting workstations have been closed three months after their initial discovery. But the reverse is true for servers; about 80% remain unresolved at the three-month mark. Domain controllers fall in the middle but actually catch up with and surpass workstations after nine months.

We could create numerous survival curves like these comparing all sorts of things, but we thought focusing on a dozen or so attributes of particular interest would be a better option. Figure 5 summarizes the effect of the selected attributes on remediation timelines.

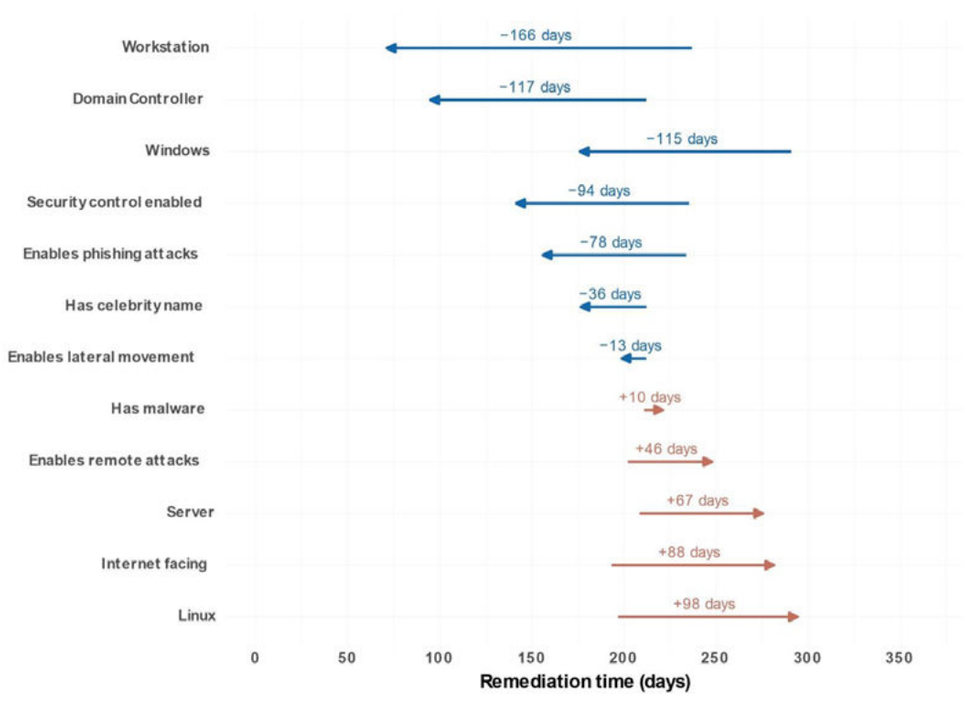


Figure 5: Effect of vulnerability attributes on average remediation timelines

We've already seen the effect for a few of the attributes in Figure 5. Workstations reduce the average remediation time by 168 days, and servers typically tack on a couple of months. It's also apparent that the operating system (OS) of the vulnerable asset makes a big difference. Microsoft Windows shaves 115 days, on average, while Linux systems take 98 days longer to remediate.

Some of the exploitation signals presented earlier also have an impact. Vulnerabilities that enable phishing and lateral movement are closed more quickly, but those that enable remote access or have been used in malware actually show longer remediation timelines. We find the latter particularly curious, as such vulnerabilities should ostensibly warrant priority remediation. The same logic applies to internet-facing assets. Based on our analysis, these traits don't directly hamper remediation. Rather, they often tend to correlate with other impeding factors (e.g., Linux servers are often internet-facing and prone to remote access).

Hoping for a view of the attack surface and remediation speed for a larger collection of vendors? If so, Figure 6 is just what you're looking for. It compares vendors based on overall vulnerability prevalence (x) and remediation speed (y). It's admittedly busy, but it effectively communicates a huge amount of useful information. We'll help you get your bearings and let you explore on your own from there.

Unlike the Gartner Magic Quadrant, you DO NOT want to be in the upper right area of this chart. Those vendors represent very common, slowly remediated vulnerabilities—which translate to a large and persistent attack surface. In the lower left, Microsoft and Google vulnerabilities are extremely common but tend to get fixed quickly. Vulnerabilities in the upper left may by flying under the radar remediation efforts because they're relatively rare and yet remain unresolved for a long time.

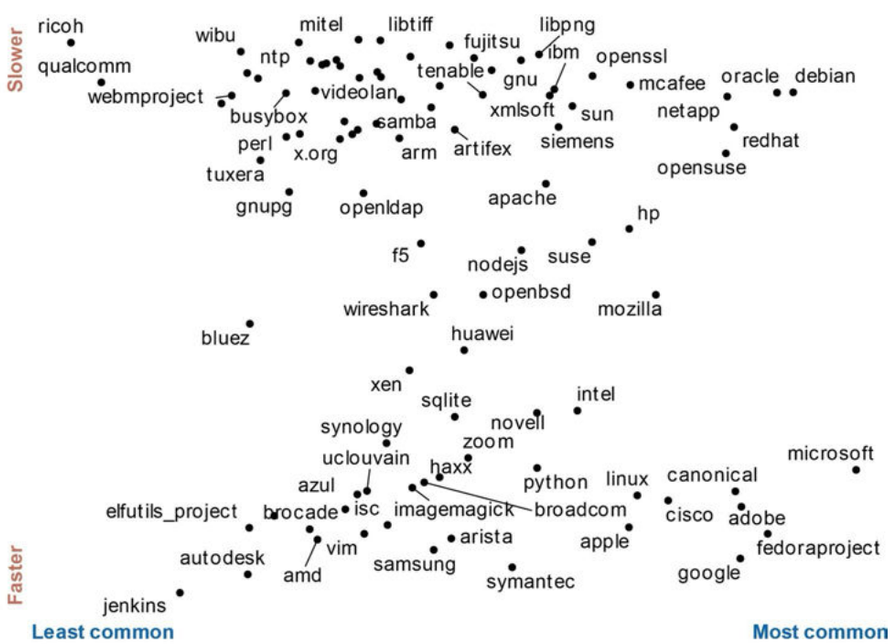


Figure 6: Comparison of vendors based on vulnerability prevalence and remediation speed

We touched on the topic of prioritizing vulnerabilities earlier, and it deserves more attention here because it's critical to the success of VM programs. It's clear that some risk factors like enabling phishing and lateral movement correlate with faster remediation, but what about overall risk scores? Do we see any evidence that they help guide prioritization and streamline remediation? Yes, we do!

Figure 7 compares remediation timelines for escalating ranges of NopSec's Business Risk Score. The green line corresponds to the highest tier of risk, and it's clear that organizations close out those vulnerabilities much faster than others. The average time-to-remediation for those is approximately two months faster than the overall average. Almost 60% of those riskiest vulnerabilities are successfully closed within three months of discovery. Compare that to the lowest risk tier (blue line)—60% of those are still around after a full year.

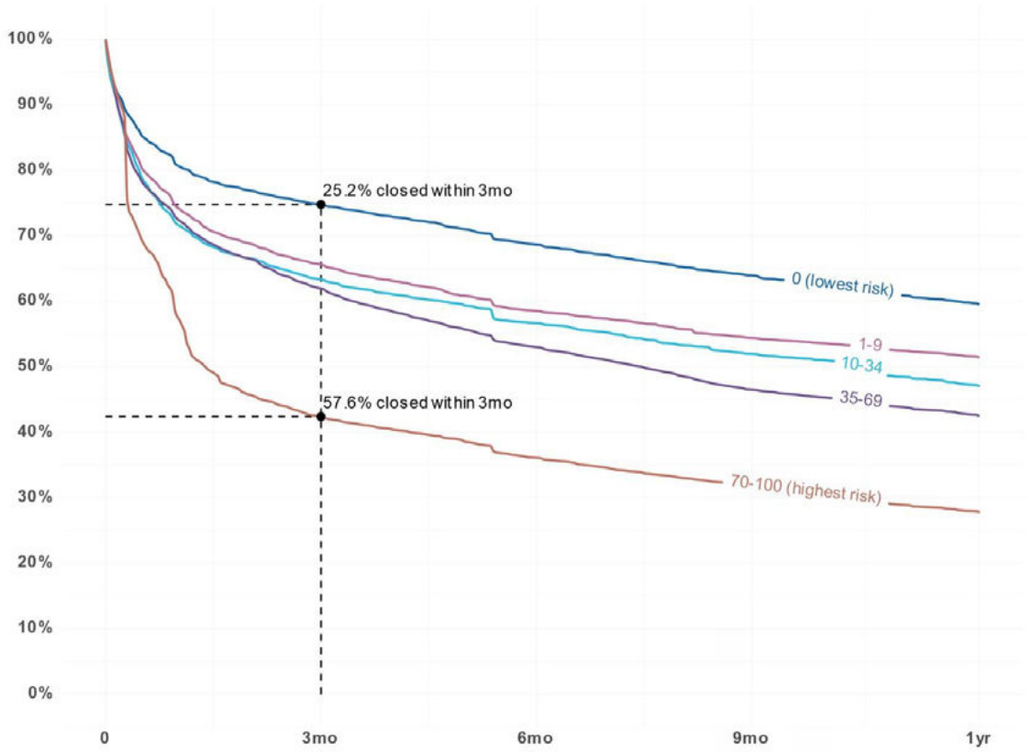


Figure 7: Vulnerability remediation timelines for NopSec's Business Risk Score ranges

Since it's impossible for organizations to fix all vulnerabilities all the time, success means knocking out the most risky ones most of the time. Figure 7 demonstrates that's not just a pie-in-the-sky aspiration. Organizations can efficiently drive down risk in the real world when following a calculated approach to prioritization.

CELEBRITY VULNERABILITIES

Earlier we saw that so-called celebrity vulns tend to be fixed faster than average. This probably has a lot to do with many of them getting scary names and logos as well as the imminent end-of-the-internet cries of doom that often accompany them. But is that a good thing? Are celebrity vulnerabilities the ones that deserve priority remediation? Do they represent the highest risk?

We created Figure 8 to help explore these important questions. Before trying to interpret it, let's get oriented to the format. All CVEs are plotted according to their assigned CVSSv3 rating on the horizontal axis and NopSec Business Risk Score on the vertical axis. Those given celebrity status are colored orange amid the field of dark gray regular 'ol vulns.

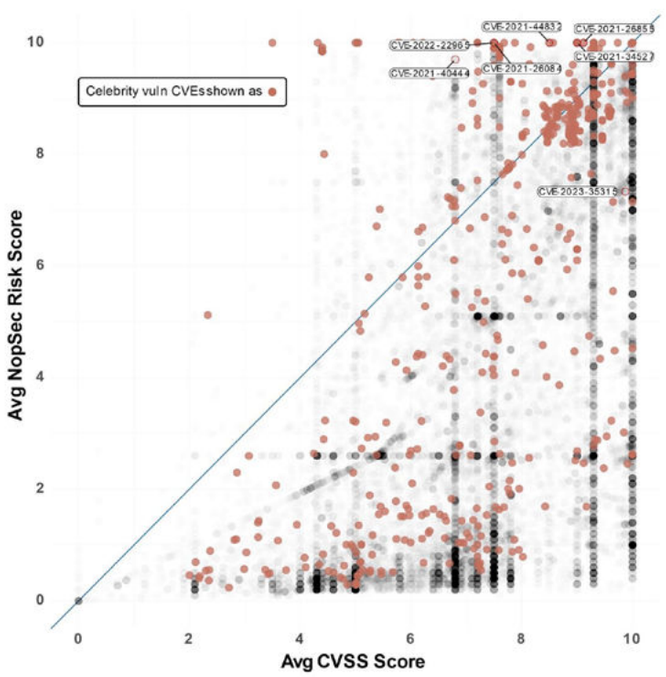


Figure 8: CVEs plotted by CVSSv3 vs. NopSec Business Risk Score with "celebrity vulns" highlighted in orange

Any CVE to the right of the diagonal line has a CVSS score that's comparatively higher than its NopSec Score. Left of the line indicates the opposite. From that, the bias of CVSS toward higher scores is immediately apparent. For VM teams, that bias leads to systemic over-prioritization and inefficient risk reduction.

Also apparent from Figure 8 is the fact that celebrity status doesn't necessarily correlate with high risk. The orange dots are all over the place—even in the portion of the chart where both CVSS and NopSec agree that vulnerabilities represent low risk. Thus, chasing celebrity vulns probably shouldn't be a key factor in your prioritization strategy.

Figure 9 offers a different but complementary perspective. It plots CVEs based on the prevalence of vulnerable assets (x-axis) and NopSec Risk Score (same as the prior chart). We did this to highlight different zones or scopes of risk. Most concerning is the upper-right quadrant, which contains high-risk vulnerabilities that affect large numbers of assets. Remediating those should top the list for prioritization to achieve maximum risk reduction.

> *While many celebrity vulns occupy the upper-right of Figure 9, the prior takeaway holds. Your prioritization strategy is better focused on risk than chasing the celebrity spotlight.*

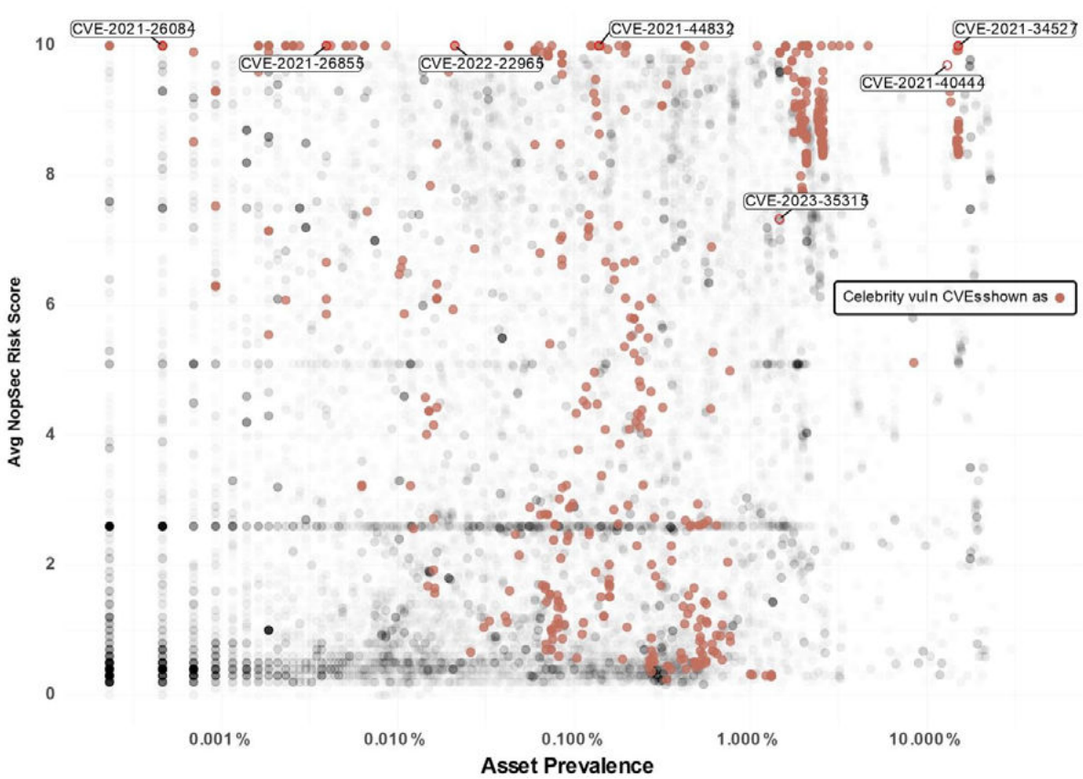


Figure 9: CVEs plotted by NopSec Business Risk Score vs. prevalence across assets with "celebrity vulns" highlighted in orange

You probably noticed that several specific CVEs are annotated in Figures 8 and 9. We did that to identify specific examples of vulnerabilities that receive a detailed analysis in the next section. This will serve to demonstrate the importance of properly assessing risk when prioritizing remediation.

VOLUME 2:

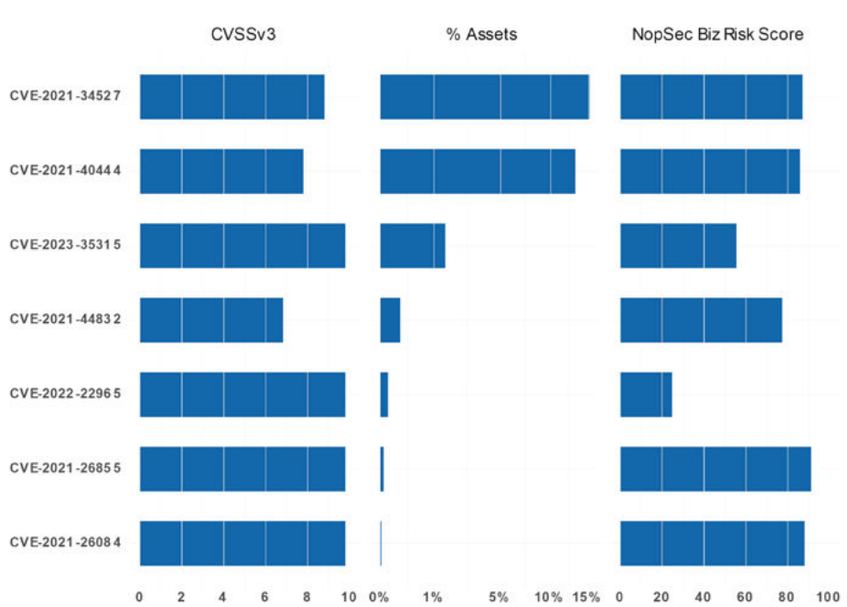
NOPSEC RISK SCORE CALCULATIONS IN REAL-LIFE SCENARIOS

THIS SECTION analyzes 36 million vulnerabilities detected by organizations using NopSec to help prioritize remediation efforts. We begin by examining the prevalence of those vulnerabilities across assets to determine which ones are most common. Then we measure how quickly those vulnerabilities are remediated and what factors speed up or slow down that process. The section closes by identifying examples of vulnerabilities that are prone to slip through the cracks of traditional prioritization strategies so your organization won't waste valuable time and energy chasing after the wrong things.

For relevance, we selected vulnerabilities for further analysis from three main categories:

- Both celebrity and non-celebrity vulnerabilities with high NopSec score and low CVSSv3 score. This category of vulnerabilities highlights the reprioritization performed by NopSec risk scoring algorithm.
- Both celebrity and non-celebrity vulnerabilities with high CVSSv3 score and low NopSec risk score. This category of vulnerabilities highlights the deprioritization performed by NopSec risk scoring algorithm.
- Both celebrity and non-celebrity vulnerabilities with high NopSec risk score present in a high percentage of assets. This category of vulnerabilities highlights the prevalence of high NopSec risk score vulnerabilities in high % of assets.

Out of these three main categories we selected a sample of seven vulnerabilities, chosen for their risk relevance and variety out of the three above mentioned categories. The selected vulnerabilities with their respective CVSSv3 score, percentage of Assets present, and NopSec Risk Score are presented in the chart below.



The seven chosen vulnerabilities will be analyzed based on the threat information present in our system at the time of writing, explaining why the selected ones were prioritized, de-prioritized and overall relevant in terms of asset prevalence. This analysis would shed light on the necessary and professional process of prioritizing security vulnerabilities as part of a modern vulnerability management program.

THE VULNERABILITY SAMPLE SELECTION

The following is the sample of vulnerabilities we selected based on the three categories highlighted above and based on their risk relevance. Below we will analyze each one of them based on their risk profile explaining why the NopSec risk score prioritized or de-prioritize them.

CVE	Vulnerability Title	NopSec Rsk Score - CVSS Score difference	% Assets	% Closed	Celebrity Vuln
CVE-2022-22965	Spring Framework JDK 9+ Remote Code Execution Vulnerability	2.50	0.0%	86.8%	Yes
CVE-2021-26084	Atlassian Confluence Server and Data Center Object-Graph Navigation Language (OGNL) Injection Vulnerability	2.50	0.0%	0.0%	Yes
CVE-2021-44832	Apache Log4j2 Remote Code Execution vulnerability	1.50	0.1%	55.4%	Yes
CVE-2021-34527	Microsoft Windows Print Spooler Remote Code Execution Vulnerability	1.00	15.1%	68.1%	Yes
CVE-2021-26855	Microsoft Exchange Server Remote Code Execution Vulnerability	0.88	0.0%	88.2%	Yes
CVE-2023-35315	Windows Layer-2 Bridge Network Driver Remote Code Execution Vulnerability	-2.54	1.5%	76.5%	No
CVE-2021-40444	Microsoft MSHTML Remote Code Execution Vulnerability	2.90	13.1%	17.6%	No

1. CVE-2022-22965 - SPRING FRAMEWORK JDK 9+ REMOTE CODE EXECUTION VULNERABILITY

A Spring MVC or Spring WebFlux application running on JDK 9+ may be vulnerable to remote code execution (RCE) via data binding. The specific exploit requires the application to run on Tomcat as a WAR deployment. If the application is deployed as a Spring Boot executable jar, i.e. the default, it is not vulnerable to the exploit. However, the nature of the vulnerability is more general, and there may be other ways to exploit it.

The vulnerability is inherently important because the Spring framework is used by several software vendors in their products, including Cisco, VMware, Siemens, and obviously Oracle.
The vulnerability is also included in the CISA KEV because the exploit is used in the wild in several active attack campaigns. The vulnerability is also identified as a celebrity vulnerability under the given name “Spring4Shell”.

The vulnerability has a CVSS score of 7.5 and a CVSSv3 score of 9.8. The vulnerability’s EPSS score is quite high at 97.48%, reflecting several exploits in the public domain, including Metasploit and several attacks observed in the wild.

Our NopSec Risk Score is at 9.263, which is 2.5 points difference to the CVSS score. This increase reflects the following feature components that made the algorithm decide for a score increase:

- Number of exploit references. Value of 9.
- Number of cyber exploit sources. Value of 803
- Number of cyber exploit sightings. Value of 2,152
- CVSSv3 score. Value of 9.8
- NLP relevant keyword: Remote code execution

The followings are the NopSec Scoe details for it to make such a score increase decision:

“nopsec_score_details”: {“is_threat”: true, “prediction”: 0.9263079762458801, “has_exploit”: true, “has_func_exp”: true}, “nopsec_threat_vectors”: {“is_malware”: true, “is_ransomware”: false, “is_exploitable”: true, “is_command_exec”: true, “is_cred_compromise”: true, “is_phishing_attack”: false, “is_priv_escalation”: true, “is_unauthenticated”: true, “is_remote_attack_scenario”: true, “is_lateral_movement_scenario”: true}, “nopsec_threat_vector_list”: [“Remote Attack Scenario”, “Known Threat”, “Allows Command Execution”, “Allows Credential Compromise”, “Known Exploit”, “Authentication Not Required”, “Allows Privilege Escalation”, “Lateral Movement Scenario”]

Therefore, we can see from this details that the vulnerability is not only a celebrity vulnerability present in the CISA KEV list but it is associated with public exploits and public exploit sightings, it is associated with malware, it is a remote code execution of an unauthenticated vulnerability which could lead to credentials compromised, privilege escalation and lateral movement scenarios. Therefore this vulnerability not only needs to be prioritized in the remediation effort but it is quite common in our clients’ organization sample and it is actively remediated.

2. CVE-2021-26084 - ATLASSIAN CONFLUENCE SERVER AND DATA CENTER OGNL INJECTION UNAUTHENTICATED REMOTE CODE EXECUTION VULNERABILITY

In affected versions of Confluence Server and Data Center, an OGNL injection vulnerability exists that would allow an unauthenticated attacker to execute arbitrary code on a Confluence Server or Data Center instance. The affected versions are before version 6.13.23, from version 6.14.0 before 7.4.11, from version 7.5.0 before 7.11.6, and from version 7.12.0 before 7.12.5.

The vulnerability is inherently important because the Atlassian Confluence platform is used in DevOps across several organizations to document business requirements and other internal documentation related to software development.

The vulnerability is also included in the CISA KEV because the exploit is used in the wild in several active attack campaigns. The vulnerability is also identified as a celebrity vulnerability.

The vulnerability has a CVSS score of 7.5 and a CVSSv3 score of 9.8. The vulnerability's EPSS score is quite high at 97.45%, reflecting several exploits in the public domain, including Metasploit and several attacks observed in the wild.

Our NopSec Risk Score is at 9.282, which is 2.5 points difference to the CVSS score. This increase reflects the following feature components that made the algorithm decide for a score increase:

- Number of exploit references. Value of 12.
- Number of cyber exploit sources. Value of 409
- Number of cyber exploit sightings. Value of 1,034
- CVSSv3 score. Value of 9.8
- Number of references on NVD. Value of 2

The followings are the NopSec Scoe details for it to make such a score increase decision:

"nopsec_score_details": {"is_threat": true, "prediction": 0.9282200932502747, "has_exploit": true, "has_func_exp": true}, "nopsec_threat_vectors": {"is_malware": true, "is_ransomware": false, "is_exploitable": true, "is_command_exec": true, "is_cred_compromise": true, "is_phishing_attack": false, "is_priv_escalation": true, "is_unauthenticated": true, "is_remote_attack_scenario": true, "is_lateral_movement_scenario": true}, "nopsec_threat_vector_list": ["Remote Attack Scenario", "Known Threat", "Allows Command Execution", "Allows Credential Compromise", "Known Exploit", "Authentication Not Required", "Allows Privilege Escalation", "Lateral Movement Scenario"]}

Therefore, we can see from this details that the vulnerability is not only a celebrity vulnerability present in the CISA KEV list but it is associated with public exploits and public exploit sightings, it is associated with malware, it is a remote code execution of an unauthenticated vulnerability which could lead to credentials compromised, privilege escalation and lateral movement scenarios. Therefore this vulnerability not only needs to be prioritized in the remediation effort but it is quite common in our clients' organization sample. However, it has not been actively remediated.

3. CVE-2021-44832 - APACHE LOG4J2 REMOTE CODE EXECUTION VULNERABILITY

Apache Log4j2 versions 2.0-beta7 through 2.17.0 (excluding security fix releases 2.3.2 and 2.12.4) are vulnerable to a remote code execution (RCE) attack when a configuration uses a JDBC Appender with a JNDI LDAP data source URI when an attacker has control of the target LDAP server. This issue is fixed by limiting JNDI data source names to the java protocol in Log4j2 versions 2.17.1, 2.12.4, and 2.3.2.

The vulnerability is inherently important because the Log4J2 library is used in several Java softwares for event logging purposes and therefore it makes the vulnerability attack surface wide spread.

The vulnerability is not currently included in the CISA KEV critical vulnerability risk. However, the vulnerability is labeled as celebrity vulnerability as it is associated with a potential exploitable element in the past (log4j) under the name “Log4Shell”.

The vulnerability has a CVSS score of 8.5 and a CVSSv3 score of 6.6. The vulnerability’s EPSS score is not that high at 2.24%, since there are not that many exploits in the public domain; however, attacks have been identified in the wild.

Our NopSec Risk Score is at 3.835, which is 1.5 points difference to the CVSS score. This increase reflects the following feature components that made the algorithm decide for a score increase:

- Presence in malware. Value of 0
- Working public exploits. Value of 0
- Number of cyber exploit sightings. Value of 0
- Number of references on NVD. Value of 2

The followings are the NopSec Scoe details for it to make such a score increase decision:

“nopsec_score_details”: {“is_threat”: false, “prediction”: 0.3835675120353699, “has_exploit”: false, “has_func_exp”: false}, “nopsec_threat_vectors”: {“is_malware”: false, “is_ransomware”: false, “is_exploitable”: false, “is_command_exec”: true, “is_cred_compromise”: true, “is_phishing_attack”: false, “is_priv_escalation”: true, “is_unauthenticated”: false, “is_remote_attack_scenario”: true, “is_lateral_movement_scenario”: true}, “nopsec_threat_vector_list”: [“Remote Attack Scenario”, “Allows Command Execution”, “Allows Credential Compromise”, “Allows Privilege Escalation”, “Lateral Movement Scenario”]}

Therefore, even though the vulnerability is listed as a celebrity vulnerability because it is associated with a vulnerable software component in the past, it is not listed in CISA KEV listing and it is not associated with working public exploits or malware / targeted attacks. No exploit sightings have been recorded. Therefore, the vulnerability’s NopSec risk score have been increased from its CVSS score but not as important as in other vulnerabilities since its exploitation probability and profile is not as prominent.

4. CVE-2021-34527 - WINDOWS PRINT SPOOLER SERVICE REMOTE CODE EXECUTION VULNERABILITY

A remote code execution vulnerability exists when the Windows Print Spooler service improperly performs privileged file operations. An attacker who successfully exploited this vulnerability could run arbitrary code with SYSTEM privileges. An attacker could then install programs; view, change, or delete data; or create new accounts with full user rights.

The vulnerability is inherently important because it allows injecting dynamic content and DLL into memory using the vulnerability exploitation. It is the perfect vulnerability to be weaponized as malware and still used in the wild by several threat actors. Several weaponized exploits and exploit frameworks are publicly available.

The vulnerability is also included in the CISA KEV because the exploit is used in the wild in several active attack campaigns. The vulnerability is also identified as a celebrity vulnerability under the name “PrintNightmare”.

The vulnerability has a CVSS score of 9.0 and a CVSSv3 score of 8.8. The vulnerability’s EPSS score is quite high at 96.74%, reflecting several exploits and exploitation frameworks in the public domain, including Metasploit and several attacks observed in the wild.

Our NopSec Risk Score is at 9.279, which is 1 point difference to the CVSS score. This increase reflects the following feature components that made the algorithm decide for a score increase:

- Number of exploit references. Value of 6.
- Number of cyber exploit sources. Value of 804
- Number of cyber exploit sightings. Value of 2,006
- CVSSv3 score. Value of 8.8
- Number of references on NVD. Value of 2

The followings are the NopSec Scoe details for it to make such a score increase decision:

“nopsec_score_details”: {“is_threat”: true, “prediction”: 0.9279255867004395, “has_exploit”: true, “has_func_exp”: true}, “nopsec_threat_vectors”: {“is_malware”: true, “is_ransomware”: false, “is_exploitable”: true, “is_command_exec”: true, “is_cred_compromise”: true, “is_phishing_attack”: false, “is_priv_escalation”: true, “is_unauthenticated”: false, “is_remote_attack_scenario”: true, “is_lateral_movement_scenario”: true}, “nopsec_threat_vector_list”: [“Remote Attack Scenario”, “Known Threat”, “Allows Command Execution”, “Allows Credential Compromise”, “Known Exploit”, “Allows Privilege Escalation”, “Lateral Movement Scenario”]}

Therefore, we can see from this details that the vulnerability is not only a celebrity vulnerability present in the CISA KEV list but it is associated with public exploits and public exploit sightings, it is associated with malware, it is a remote code execution of an unauthenticated vulnerability which could lead to credentials compromised, privilege escalation and lateral movement scenarios. Therefore this vulnerability not only needs to be prioritized in the remediation effort but it is quite common in our clients’ organization sample. At the same time, it has been actively remediated.

5. CVE-2021-26855 - MICROSOFT EXCHANGE SERVER REMOTE CODE EXECUTION VULNERABILITY

Vulnerability on Microsoft Exchange Server allows an attacker bypassing the authentication, impersonating as the admin (CVE-2021-26855) and writing an arbitrary file (CVE-2021-27065) to get the RCE (Remote Code Execution).

The vulnerability is inherently important because it allows an unauthenticated attacker to either impersonate an admin or write arbitrary files to get remote code execution with SYSTEM privileges. It is the perfect vulnerability that was weaponized to create self-spreading malware.

The vulnerability is also included in the CISA KEV because the exploit is used in the wild in several active attack campaigns. The vulnerability is also identified as a celebrity vulnerability under the name “ProxyLogon”.

The vulnerability has a CVSS score of 7.5 and a CVSSv3 score of 9.8. The vulnerability’s EPSS score is quite high at 97.48%, reflecting several exploits and exploitation frameworks in the public domain, including Metasploit and several attacks observed in the wild.

Our NopSec Risk Score is at 9.275, which is 0.88 point difference to the CVSS score. This increase reflects the following feature components that made the algorithm decide for a score increase:

- Number of exploit references. Value of 23.
- Number of cyber exploit sources. Value of 874
- Number of cyber exploit sightings. Value of 3,364
- CVSSv3 score. Value of 9.1

The followings are the NopSec Scoe details for it to make such a score increase decision:

“nopsec_score_details”: {“is_threat”: true, “prediction”: 0.9275614619255066, “has_exploit”: true, “has_func_exp”: true}, “nopsec_threat_vectors”: {“is_malware”: true, “is_ransomware”: false, “is_exploitable”: true, “is_command_exec”: true, “is_cred_compromise”: true, “is_phishing_attack”: false, “is_priv_escalation”: true, “is_unauthenticated”: true, “is_remote_attack_scenario”: true, “is_lateral_movement_scenario”: true}, “nopsec_threat_vector_list”: [“Remote Attack Scenario”, “Known Threat”, “Allows Command Execution”, “Allows Credential Compromise”, “Known Exploit”, “Authentication Not Required”, “Allows Privilege Escalation”, “Lateral Movement Scenario”]}

Therefore, we can see from these details that the vulnerability is not only a celebrity vulnerability present in the CISA KEV list but it is associated with public exploits and public exploit sightings, it is associated with malware, it is a remote code execution of an unauthenticated vulnerability which could lead to credentials compromised, privilege escalation and lateral movement scenarios. Therefore this vulnerability not only needs to be prioritized in the remediation effort but it is quite common in our clients’ organization sample. At the same time, it has been actively remediated.

6. CVE-2023-35315 - WINDOWS LAYER-2 BRIDGE NETWORK DRIVER REMOTE CODE EXECUTION VULNERABILITY

An unauthenticated attacker could exploit the vulnerability by sending a specially crafted request to a Windows Server configured as a Layer-2 Bridge.

The vulnerability has been scored as 8.8 in the CVSSv3 scale. However, the vulnerability can be easily deprioritized because it does not have any public exploit code available and it is not exploited in the wild.

The vulnerability is not included in the CISA KEV listing since it is a remote code execution with difficult exploitation and exploit code not made public.

The vulnerability has a CVSSv3 score of 8.8. The vulnerability's EPSS score is very low at 0.13%, since public exploits and exploitation in the wild is not present.

Our NopSec Risk Score is at 3.073, which is 2.54 points less than the CVSS score. This decrease reflects the following feature components that made the algorithm decide for a score decrease:

- Number of exploit references. Value of 23.
- Number of cyber exploit sources. Value of 3
- Number of cyber exploit sightings. Value of 5. Low number
- CVSSv3 score. Value of 8.8

The followings are the NopSec Score details for it to make such a score increase decision:

"nopsec_score_details": {"is_threat": false, "prediction": 0.3073217272758484, "has_exploit": true, "has_func_exp": false}, "nopsec_threat_vectors": {"is_malware": false, "is_ransomware": false, "is_exploitable": false, "is_command_exec": true, "is_cred_compromise": true, "is_phishing_attack": false, "is_priv_escalation": true, "is_unauthenticated": true, "is_remote_attack_scenario": true, "is_lateral_movement_scenario": true}, "nopsec_threat_vector_list": ["Remote Attack Scenario", "Allows Command Execution", "Allows Credential Compromise", "Authentication Not Required", "Allows Privilege Escalation", "Lateral Movement Scenario"]}

Therefore, we can see from these details that the vulnerability is not a celebrity vulnerability and it is not included in the CISA KEV listing. It is not associated with public exploit code and exploitation in the wild. Therefore the vulnerability has been deprioritized by the NopSec risk score.

7. CVE-2021-40444 - MICROSOFT MSHTML REMOTE CODE EXECUTION VULNERABILITY

Microsoft is investigating reports of a remote code execution vulnerability in MSHTML that affects Microsoft Windows. Microsoft is aware of targeted attacks that attempt to exploit this vulnerability by using specially-crafted Microsoft Office documents. An attacker could craft a malicious ActiveX control to be used by a Microsoft Office document that hosts the browser rendering engine. The attacker would then have to convince the user to open the malicious document. Users whose accounts are configured to have fewer user rights on the system could be less impacted than users who operate with administrative user rights.

The vulnerability is not included in the CISA KEV listing and it is not a celebrity vulnerability. However, the vulnerability has exploit code available and it has been leveraged as part of several ransomware campaigns.

The vulnerability has a CVSS score of 6.8 and a CVSSv3 score of 7.8. The vulnerability's EPSS score is quite high at 96.82%, reflecting several exploits in the public domain, including Metasploit and several attacks observed in the wild.

Our NopSec Risk Score is at 9.278, which is 2.9 points difference to the CVSS score. This increase reflects the following feature components that made the algorithm decide for a score increase:

- Number of exploit references. Value of 10.
- Number of cyber exploit sources. Value of 1,282.
- Number of cyber exploit sightings. Value of 4,390.
- CVSSv3 score. Value of 8.8

The followings are the NopSec Scoe details for it to make such a score increase decision:

"nopsec_score_details": {"is_threat": true, "prediction": 0.9278809428215027, "has_exploit": true, "has_func_exp": true}, "nopsec_threat_vectors": {"is_malware": true, "is_ransomware": false, "is_exploitable": true, "is_command_exec": true, "is_cred_compromise": true, "is_phishing_attack": true, "is_priv_escalation": true, "is_unauthenticated": true, "is_remote_attack_scenario": true, "is_lateral_movement_scenario": true}, "nopsec_threat_vector_list": ["Remote Attack Scenario", "Known Threat", "Phishing Attack", "Allows Command Execution", "Allows Credential Compromise", "Known Exploit", "Authentication Not Required", "Allows Privilege Escalation", "Lateral Movement Scenario"]}

Therefore, the vulnerability is not a celebrity vulnerability and is not included in the CISA KEV but it is a MS Word vulnerability that is easily exploitable as part of social engineering attacks, it has been utilized as part of ransomware campaigns, it has public exploit available also included in the Metasploit framework. Active exploitation in the wild made it the top chart in terms of reprioritization of exploitation risk.

Through the vulnerability examples above, we have seen that vulnerabilities - either present or not in the CISA KEV list and being considered celebrity vulnerabilities - can be prioritized or deprioritized based on threat intelligence sightings, inclusion into malware, being used as part of attacks in the wild, having easily usable public exploits and having remote code execution characteristics. Those features help the algorithm prioritize the remedial actions even if malware exploitation or attack in the wild is not currently taking place.

VOLUME 3:

CYBER THREAT & EXPOSURE MANAGEMENT

MAPPING MITRE ATT&CK TAXONOMIES TO CVES

The MITRE ATT&CK framework is a knowledge base of adversary tactics, techniques, and procedures (TTPs) based on real-world observations of attacks. The framework is subdivided into tactics, which correlate to the stages in the lifecycle of an attack. Each tactic is associated with a number of different techniques that are further divided into sub-techniques associated with procedures (tools/malware), mitigating controls, and detection strategies. The ATT&CK guide is highly granular and provides asset owners with meaningful contextual details about an attacker's exploitation path through their environment. For example, it postulates that an attacker has exploited an information disclosure vulnerability on a web application server that resulted in unauthorized access to session cookies. The ATT&CK knowledge base could provide insights into the tactics and techniques an attacker would use next to further his foothold in the exploited environment.

Established in the 1960s MITRE is a familiar name in the cyber security space, having aided in the development of numerous standards including Common Vulnerabilities and Exposures (CVE), Common Platform Enumeration (CPE), and Common Weakness Enumeration (CWE). These related classification standards provide system owners with information about the risk impacting their environments. For instance, CPE helps identify a system's specific product and version, CVE informs what vulnerabilities that specific product and version is exposed to, and CWE provides insight into the root cause of the vulnerabilities. These standards are bedrocks of vulnerability and exposure classification and embraced by the cyber security community.

Despite being hatched from the same parent entity, the ATT&CK framework exists independently from CPE, CVE, and CWE. The lack of a direct mapping between the CVE and ATT&CK taxonomies leaves a gap in the contextual understanding of the overall security risk. In an ideal world, system owners could easily query all CVEs in their environment that could result in lateral movement (ATT&CK Tactic TA0008). However this is presently not possible using the available MITRE risk classification systems.

> Successfully correlating a vulnerability (CVE) to an ATT&CK TTP requires a subject matter expert that understands how an attacker identifies a vulnerability, how an attacker exploits a vulnerability, and the potential impact of successful exploitation. The confluence of expert knowledge and semantics enables the logical mapping of a CVE to the most relevant ATT&CK taxonomies. Without expert knowledge, accurately mapping a CVE to a related ATT&CK technique presents challenges. In the absence of expert knowledge NopSec believes that defining novel relationships between CVE and ATT&CK taxonomies is a problem well suited for large language models (LLMs).

ATT&CK FRAMEWORK AND LLMS

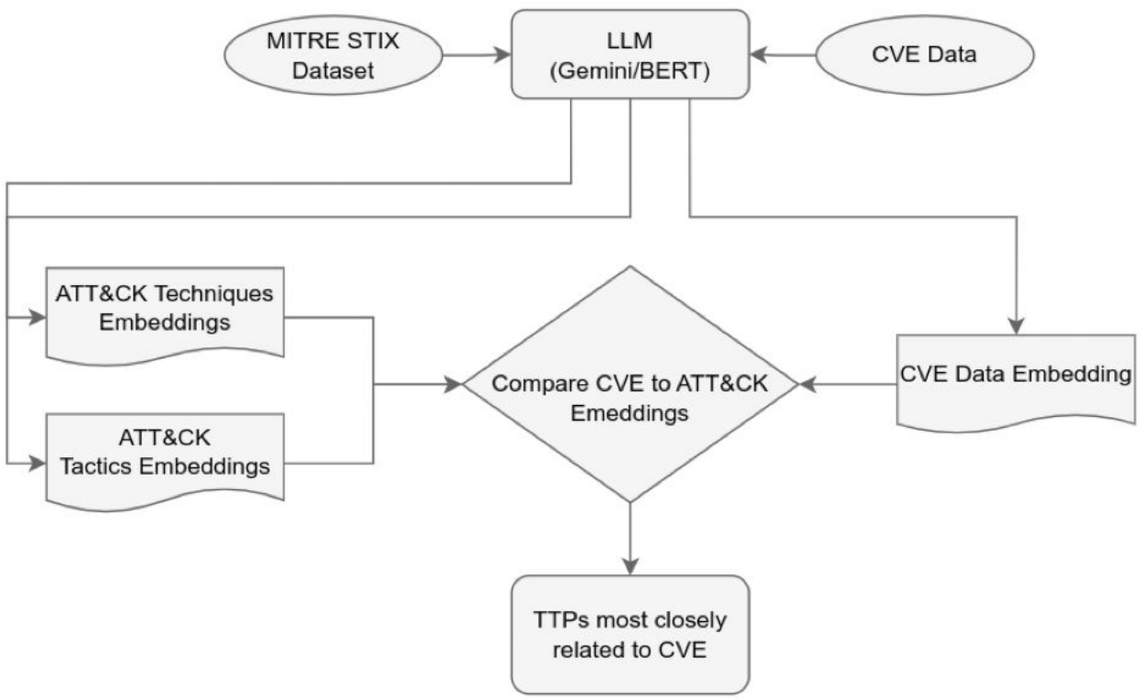
An LLM (Large Language Model) is a form of artificial intelligence that is used in general purpose language generation. LLMs achieve this by calculating a model of statistical relationships between large quantities of text. LLMs are an emergent, still evolving technology, but their presence is nearly ubiquitous in the form of internet search engines and generative AI such as ChatGPT and Google Gemini LLM engines. In the context of LLMs, like Gemini, it's possible to extract a desired output by crafting a prompt. A prompt can be used to generate data or establish relationships between unstructured text. For example, a natural language prompt could be crafted to compel a LLM to generate a resume or compose a blog article. This is possible because LLMs are trained on the syntax, semantics, and ontology of language. The same cognitive abilities applied by a subject matter expert when analyzing and understanding the nature of a CVE are approximated at massive scales and incredible speeds using a LLM.

NopSec endeavored to leverage LLMs as a means to correlate CVE details, such as those available via the National Vulnerability Database (NVD), to specific MITRE ATT&CK tactics and techniques. The objective of this exercise is to understand for each CVE which ATT&CK adversary TTPs (tactics, techniques and procedures) an attacker could choose to further consolidate his foothold in the exploited environment. There exists a wide variety of models trained to satisfy different needs. Models such as Gemini and GPT are multimodal, capable of processing a wide spectrum of input including text, audio, and images. Other models, called unimodal, are trained on a specific dataset to specialize in general language around a specific topic, such as computer programming. At their core, LLMs are not what they were trained on, but what kind of transformer architecture was used to vectorize input such as encoder-decoder, encoder-only, or decoder-only. For the purpose of our research, NopSec focused on encoder-only and decoder-only models.

Generally speaking, encoder-only models specialize in extracting contextual information from a tokenized input sequence and are well suited to establishing semantic relationships and context. A decoder-only model specializes in generating a coherent sequence of tokens based on contextual information. They predict the next token based on the context of the previous one. Encoder-only provides context, decoder-only generates related sequences of tokens, but in every case input is tokenized and embedded into the model.

NopSec elected to use two different models to assist in mapping CVEs to ATT&CK techniques: Google Gemini, a multimodal LLM that leverages decoder-only transformer architecture and Bidirectional Encoder Representations from Transformers (BERT) a unimodal, encoder-only LLM. These two models served as the basis for NopSec CVE to ATT&CK mapping research. The BERT based ATTACK LLM was fine tuned on cybersecurity domain-specific input data from the MITRE ATT&CK framework, which is in contrast to the much deeper and expansive training set of Google Gemini.

NopSec used each LLM to encode a matrix of embeddings calculated from MITRE enterprise ATT&CK data, which contained information about the TTPs that comprise the ATT&CK framework. Publicly available CVE data provided by National Vulnerability Database (NVD) was encoded and vectorized using the LLMs. Using this group of embeddings it was possible to determine the degree to which the input CVE data was related to the ATT&CK TTPs. The diagram below provides an illustration of NopSec CVE to ATT&CK mapping workflow.



Using this test harness NopSec was able to categorize CVEs into their related TTPs. NopSec began the evaluation by creating two (2) distinct embeddings.

- **ATT&CK data by attack technique;**
- **ATT&CK data by attack tactic and technique**

MAPPING CVE BY ATT&CK TECHNIQUE

The first embedding was based purely on ATT&CK technique information, which consisted of the technique title and detailed description, indexed by the TTP ID. The data was encoded using the Gemini and BERT LLMs and vectorized to generate the embeddings. The ATT&CK techniques embedding could be used to identify similarities between embedded CVE data and the ATT&CK framework. The tables below contain the results of this method applied to the previously selected group of CVEs.

CVE	Vulnerability	Technique ID	Technique Title	Confidence
CVE-2021-26084	Atlassian Confluence Server and Data Center Object-Graph Navigation Language (OGNL) Injection Vulnerability	T1221	Template Injection	0.455416143
		T1190	Exploit Public-Facing Application	0.403412282
CVE-2021-26855	Microsoft Exchange Server Remote Code Execution Vulnerability	T1203	Exploitation for Client Execution	0.5792467
		T1211	Exploitation for Defense Evasion	0.487958282
CVE-2021-34527	Microsoft Windows Print Spooler Remote Code Execution Vulnerability	T1068	Exploitation for Privilege Escalation	0.559049904
		T1222	File and Directory Permissions Modification	0.554395974
CVE-2021-40444	Microsoft MSHTML Remote Code Execution Vulnerability	T1203	Exploitation for Client Execution	0.645931482
		T1211	Exploitation for Defense Evasion	0.54076308
CVE-2021-44832	Apache Log4j2 Remote Code Execution vulnerability	T1654	Log Enumeration	0.335945278
		T1558	Steal or Forge Kerberos Tickets	0.288426935
CVE-2022-22965	Spring Framework JDK 9+ Remote Code Execution Vulnerability	T1203	Exploitation for Client Execution	0.378445506
		T1505	Server Software Component	0.37290591
CVE-2023-35315	Windows Layer-2 Bridge Network Driver Remote Code Execution Vulnerability	T1599	Network Boundary Bridging	0.485672235
		T1047	Windows Management Instrumentation	0.452629626

CVE mapping based on BERT ATT&CK technique embedding with sub-techniques excluded

CVE	Vulnerability	Technique ID	Technique Title	Confidence
CVE-2021-26084	Atlassian Confluence Server and Data Center Object-Graph Navigation Language (OGNL) Injection Vulnerability	T1221	Template Injection	0.618033445
		T1072	Software Deployment Tools	0.597603526
CVE-2021-26855	Microsoft Exchange Server Remote Code Execution Vulnerability	T1212	Exploitation for Credential Access	0.650463668
		T1210	Exploitation of Remote Services	0.649826794
CVE-2021-34527	Microsoft Windows Print Spooler Remote Code Execution Vulnerability	T1210	Exploitation of Remote Services	0.65553795
		T1547	Boot or Logon Autostart Execution	0.64199945
CVE-2021-40444	Microsoft MSHTML Remote Code Execution Vulnerability	T1220	XSL Script Processing	0.715400308
		T1203	Exploitation for Client Execution	0.703556555
CVE-2021-44832	Apache Log4j2 Remote Code Execution vulnerability	T1059	Command and Scripting Interpreter	0.60849264
		T1207	Rogue Domain Controller	0.601446789
CVE-2022-22965	Spring Framework JDK 9+ Remote Code Execution Vulnerability	T1203	Exploitation for Client Execution	0.614510523
		T1190	Exploit Public-Facing Application	0.607854304
CVE-2023-35315	Windows Layer-2 Bridge Network Driver Remote Code Execution Vulnerability	T1652	Device Driver Discovery	0.625530751
		T1210	Exploitation of Remote Services	0.611468556

CVE mapping based on Gemini ATT&CK technique embedding with sub-techniques excluded

As a means to vet the accuracy of the above results NopSec manually conducted the same exercise and mapped CVEs to two (2) ATT&CK TTPs, leveraging only the NVD provided CVE description to evaluate the vulnerability. The table below captures the results of the manual analysis as well as the BERT and Gemini output, together with the two models' accuracy percentage compared to the manual mapping.

CVE	Vulnerability	Subject Matter Expert Mapping	BERT Mapping	Gemini Mapping	BERT vs Manual	Gemini vs Manual
CVE-2021-26084	Atlassian Confluence Server and Data Center Object-Graph Navigation Language (OGNL) Injection Vulnerability	Exploit Public-Facing Application (T1190)	Template Injection (T1221)	Command and Scripting Interpreter (T1059)	0%	50%
		Data Manipulation (T1565)	Software Deployment Tools (T1072)	Exploitation of Remote Services (T210)		

CVE-2021-26855	Microsoft Exchange Server Remote Code Execution Vulnerability	Command and Scripting Interpreter (T1059)	Command and Scripting Interpreter (T1059)	Command and Scripting Interpreter (T1059)	100%	100%
		Exploitation of Remote Services (T1210)	Exploitation of Remote Services (T1210)	Exploitation of Remote Services (T1210)		
CVE-2021-34527	Microsoft Windows Print Spooler Remote Code Execution Vulnerability	Exploitation for Privilege Escalation (T1068)	Exploitation for Privilege Escalation (T1068)	Exploitation of Remote Services (T1210)	50%	0%
		Exploitation of Remote Services (T1210)	File and Directory Permissions Modification (T1222)	Boot or Logon Autostart Execution (T1547)		
CVE-2021-40444	Microsoft MSHTML Remote Code Execution Vulnerability	User Execution (T1204)	Exploitation for Client Execution (T1203)	XSL Script Processing (T1220)	50%	50%
		Exploitation of Remote Services (T1210)	Exploitation for Defense Evasion (T1211)	Exploitation for Client Execution (T1203)		
CVE-2021-44832	Apache Log4j2 Remote Code Execution vulnerability	Exploitation of Remote Services (T1210)	Log Enumeration (T1654)	Command and Scripting Interpreter (T1059)	0%	0%
		Exploit Public-Facing Application (T1190)	Steal or Forge Kerberos Tickets (T1558)	Rogue Domain Controller (T1207)		
CVE-2022-22965	Spring Framework JDK 9+ Remote Code Execution Vulnerability	Exploit Public-Facing Application (T1190)	Exploitation for Client Execution (T1203)	Exploitation for Client Execution (T1203)	50%	50%
		Server Software Component (T1505)	Server Software Component (T1505)	Exploit Public-Facing Application (T1190)		
CVE-2023-35315	Windows Layer-2 Bridge Network Driver Remote Code Execution Vulnerability	Device Driver Discovery (T1652)	Network Boundary Bridging (T1599)	Device Driver Discovery (T1652)	0%	50%
		Exploitation of Remote Services (T1210)	Windows Management Instrumentation (T1047)	Exploitation of Remote Services (T1210)		
Legend:				**Overall Accuracy**	35.7%	42.9%
		TTP Exact match				
		TTP Plausibly related				
		TTP Unrelated				

Manually mapped CVEs compared against Gemini and BERT ATT&CK technique embedding

NopSec found that Google Gemini was the engine that was most closely aligned with manually mapped TTPs when mapping a CVE to a related ATT&CK technique based purely on an NVD description of the vulnerability and technique description. This exercise highlighted the limitations of correlating a CVE to relevant ATT&CK TTPs based purely on the CVE vulnerability description. Within NVD there exists a great deal of variance in the details present in a vulnerability description. Some vulnerabilities, such as CVE-2021-26855, contained very little information to key off, “Microsoft Exchange Server Remote Code Execution Vulnerability.” This impacted the ability of an LLM or human analyst to arrive at a meaningful TTP. And this is inherently a limitation of language used in the formulation of CVE description, not the LLM’s capability to correlate single word tokens. Interestingly, due to the limited data available it was the only vulnerability where BERT, Gemini, and a human analyst were in complete agreement. Although the results indicated that the multimodal Gemini LLM was more accurate the difference was negligible. It should be noted that some of the results returned by BERT and Gemini were plausible and could be considered as applicable to the vulnerability described by the CVE.

WHERE DO WE GO FROM HERE?

LLMs are fascinating and powerful tools. In our use case they enabled a person to analyze an extraordinary quantity of CVEs in a short amount of time, however this was achieved at the cost of accuracy. This is a general rule of thumb for any LLM. The output should be taken with a grain of salt.

That said, one of the more interesting takeaways from the research was that the LLMs identified relevant TTPs that were missed by a human analyst. The volume of data present in the MITRE ATT&CK framework is vast, which makes an exhaustive manual search impractical. A subject matter expert can isolate tactics most likely related to a TTP, but this is no guarantee that all relevant TTPs will be identified. The fact that Gemini and BERT respectively achieved 43% and 37% alignment to a human analyst is fairly remarkable.

It’s clear from the results of our research that LLMs can complement our understanding of risk by providing relevant contextual information about the nature of a vulnerability. Regardless of the LLM, we can conclude that the inclusion of more information in the query/prompt increases the accuracy of the CVE to TTP mapping.

When the CVE input data was limited to a single sentence the accuracy of the results decreased significantly. The same is true of the embedded MITRE ATT&CK data. When leveraging the ATT&CK tactic description first and ATT&CK technique description as a secondary match we observed a sharp decrease in accuracy. In each case truncating the embedded data resulted in decreased accuracy and relevance of the results. It is however possible to have an excess of detail. Not included in this report was any mention of MITRE ATT&CK sub-techniques. The ATT&CK framework is arranged hierarchically with tactics at the root level followed by techniques, and sub-techniques. Sub-techniques were initially included as part of the analysis, but were eventually eliminated due to the excess quantity of invalid results. Also excluded from our analysis were the MITRE ATT&CK Data Sources. The Data Sources represent the data an attacker wants to target, such as Active Directory, or the original source of an attack, such as Network Traffic. The Data Source component of the ATT&CK framework contains information that would prove valuable in furthering our contextual understanding of risk as well as increasing the overall accuracy of any LLM derived means of establishing textual relationships. However, it was out of scope for the purpose of this research.

Finally, our research highlighted the highly subjective nature of mapping CVEs to TTPs. The ATT&CK TTPs describe what attack paths an attacker may leverage and how the paths are executed. In the context of a vulnerability that results in remote command execution, an attack path could include a significant number of related TTPs. Once remote command execution is achieved there are few barriers to limit other attacks, thus more TTPs become plausible as a link in the attack chain. However, not all of the TTPs have equal probability of being part of a real-world attack path, despite being possible. These contextual nuances result in inconsistent TTP relationships even when evaluated by a human analyst. No two analysts will arrive at an identical attack path. This fundamental reality makes evaluating the accuracy of an LLM derived mapping subjective.

WHY DOES THIS MATTER?

The techniques described above highlight the importance of mapping the vulnerability information that are first exploited by an attacker to the ATT&CK TTPs that are further developed down the attack chains to further deepen the foothold that an attacker has on the exploited environment. In other words, which vulnerability is exploited first as a starting point could modify entirely the exploited attack chain.

As a demonstration to this, it is worth describing the selected CVEs and related ATT&CK TTPs to explain the entire attack chain.

For example, for CVE-2021-34527, the "Print Nightmare" celebrity vulnerability which involves

"A remote code execution vulnerability when the Windows Print Spooler service improperly performs privileged file operations. An attacker who successfully exploited this vulnerability could run arbitrary code with SYSTEM privileges. An attacker could then install programs; view, change, or delete data; or create new accounts with full user rights".

The BERT LLM model correctly maps this vulnerability to the TTP "Event Triggered Execution (T1546)", which means that the vulnerability (RCE) allows the execution of a payload to be executed on disk or in memory.

The Gemini LLM correctly maps this vulnerability to the TTPs "Boot or Logon Autostart Execution (T1547)" and "Scheduled Task/Job (T1053)", which are two lateral movement techniques to both install self-executing tasks at Logon or Boot using the Remote Code Execution for task instantiation and to schedule and execute tasks and jobs to enable lateral movement at a later time. Again, the vulnerability is the enabler of the subsequent selected TTPs techniques and tactics.

In another example of CVE-2021-40444, a Microsoft Windows MSHTML Remote Code Execution:

"A remote code execution vulnerability exists in MSHTML that affects Microsoft Windows. Microsoft is aware of targeted attacks that attempt to exploit this vulnerability by using specially-crafted Microsoft Office documents. An attacker could craft a malicious ActiveX control to be used by a Microsoft Office document that hosts the browser rendering engine. The attacker would then have to convince the user to open the malicious document. Users whose accounts are configured to have fewer user rights on the system could be less impacted than users who operate with administrative user rights."

Both BERT and Gemini models correctly map this vulnerability with Exploitation for Client Execution (T1203) ATT&CK TTP which points to the subsequent exploitation of client-side software components, in this case Microsoft Office components.

Even for XSL Script processing (T1220) which is mapped by Gemini model, this technique is often used by attackers to bypass controls in the application. From the ATT&CK framework description:

"Adversaries may bypass application control and obscure execution of code by embedding scripts inside XSL files. Extensible Stylesheet Language (XSL) files are commonly used to describe the processing and rendering of data within XML files. To support complex operations, the XSL standard includes support for embedded scripting in various languages. [1]

Adversaries may abuse this functionality to execute arbitrary files while potentially bypassing application control. Similar to Trusted Developer Utilities Proxy Execution, the Microsoft common line transformation utility binary (msxsl.exe) [2] can be installed and used to execute malicious JavaScript embedded within local or remote (URL referenced) XSL files. [3] Since msxsl.exe is not installed by default, an adversary will likely need to package it with dropped files. [4] Msxsl.exe takes two main arguments, an XML source file and an XSL stylesheet. Since the XSL file is valid XML, the adversary may call the same XSL file twice. When using msxsl.exe adversaries may also give the XML/XSL files an arbitrary file extension.[5]"

Therefore, based on these examples, the interpretation of the mapped TTPs to the CVE by the LLM models makes total sense if we consider the TTPs "extensions and continuations" of the attack paths after the initial vulnerabilities are exploited. And that is the reason why it is important mapping each CVE to the related lateral movement techniques exposed by the ATT&CK TTPs.

CONCLUSIONS

At the onset of our research we were unsure if our approach would bear meaningful results. Having established a viable method to autonomously map a CVE to attack taxonomies, we are excited to improve the accuracy of the output. Relying on a CVE description and title alone provides insufficient information to consistently map a CVE to related TTPs. Richer contextual information about the vulnerability is required to ensure that embedded queries are more complete and thus better suited to providing accurate results. To this end NopSec intends to leverage their proprietary threat intelligence feed to augment NVD CVE information. The increased richness of the CVE query will be combined with thoughtful query tokenization to reduce misleading information that may confuse an LLM. We're just scratching the surface of the positive impact LLMs can have on our understanding of cyber risk. By understanding the attack taxonomies of CVEs it becomes possible to establish complete attack chains. Our goal is to introduce all these features into our flagship Attack Surface Management solution. We think like hackers. We think you should too.

The table below summarizes the results of our analysis exposing the most likely mapping between selected CVEs and ATT&CK TTPs.

CVE	Vulnerability Title	NopSec Rsk Score - CVSSv3 Score difference	Selected ATT&CK TTPs
CVE-2022-22965	**Spring Framework JDK 9+ Remote Code Execution Vulnerability**	+2.50	Exploit Public-Facing Application (T1190)
			Server Software Component (T1505)
CVE-2021-26084	**Atlassian Confluence Server and Data Center Object-Graph Navigation Language (OGNL) Injection Vulnerability**	+2.50	Command and Scripting Interpreter (T1059)
			Exploit Public-Facing Application (T1190)
CVE-2021-44832	**Apache Log4j2 Remote Code Execution vulnerability**	+1.50	Command and Scripting Interpreter (T1059)
			Exploit Public-Facing Application (T1190)
CVE-2021-34527	**Microsoft Windows Print Spooler Remote Code Execution Vulnerability**	+1.00	Exploitation for Privilege Escalation (T1068)
			Exploitation of Remote Services (T1210)
CVE-2021-26855	**Microsoft Exchange Server Remote Code Execution Vulnerability**	+0.88	Command and Scripting Interpreter (T1059)
CVE-2023-35315	**Windows Layer-2 Bridge Network Driver Remote Code Execution Vulnerability**	-2.54	Exploitation of Remote Services (T1210) Device Driver Discovery (T1652)
			Exploitation of Remote Services (T1210)
CVE-2021-40444	**Microsoft MSHTML Remote Code Execution Vulnerability**	+2.90	User Execution (T1204)
			Exploitation of Remote Services (T1210)

REFERENCES

- https://huggingface.co/basel/ATTACK-BERT
- https://ai.google.dev/gemini-api/docs?_gl=1*16urg7e*_up*MQ..&gclid=CjwKCAjw1K-zBhBIEiwAWeCOF50Bb42t0z0-xWZN4n6hASycfUUUF1VwKpxmSw4XJkKCxkhtXe3EFBoCnsQQAvD_BwE
- https://en.wikipedia.org/wiki/BERT_(language_model)

APPENDIX

APPENDIX A

Mapping CVE by ATT&CK Tactic and Technique

NopSec next generated an embedding of ATT&CK techniques organized by tactic. This solution limits results to a subset of ATT&CK techniques that fall under related tactics. The CVE text is first compared against ATT&CK tactic descriptions to filter unrelated tactics. The CVE text is then compared against the techniques that are associated with each tactic. This method of identifying relevant TTPs should reduce the number of unrelated techniques by filtering irrelevant tactics. The tables below contain the results of this method applied to the previously selected group of CVEs.

CVE	Vulnerability	Tactic ID	Tactic Title	Technique ID	Technique	Confidence
CVE-2021-26084	Atlassian Confluence Server and Data Center Object-Graph Navigation Language (OGNL) Injection Vulnerability	TA0002	Execution	T1203	Exploitation for Client Execution	0.413596898
		TA0040	Impact	T1565	Data Manipulation	0.286306411
CVE-2021-26855	Microsoft Exchange Server Remote Code Execution Vulnerability	TA0002	Execution	T1203	Exploitation for Client Execution	0.603139341
		TA0005	Defense Evasion	T1211	Exploitation for Defense Evasion	0.54715991
CVE-2021-34527	Microsoft Windows Print Spooler Remote Code Execution Vulnerability	TA0004	Privilege Escalation	T1546	Event Triggered Execution	0.55662322
		TA0003	Persistence	T1546	Event Triggered Execution	0.55662322
CVE-2021-40444	Microsoft MSHTML Remote Code Execution Vulnerability	TA0002	Execution	T1203	Exploitation for Client Execution	0.655006945
		TA0005	Defense Evasion	T1211	Exploitation for Defense Evasion	0.581152081
CVE-2021-44832	Apache Log4j2 Remote Code Execution vulnerability	TA0006	Credential Access	T1558	Steal or Forge Kerberos Tickets	0.284517944
		TA0006	Credential Access	T1212	Exploitation for Credential Access	0.256812423
CVE-2022-22965	Spring Framework JDK 9+ Remote Code Execution Vulnerability	TA0002	Execution	T1203	Exploitation for Client Execution	0.381921828
		TA0004	Privilege Escalation	T1068	Exploitation for Privilege Escalation	0.33768484
CVE-2023-35315	Windows Layer-2 Bridge Network Driver Remote Code Execution Vulnerability	TA0005	Defense Evasion	T1599	Network Boundary Bridging	0.473288685
		TA0002	Execution	T1047	Windows Management Instrumentation	0.469915897

CVE mapping based on BERT ATT&CK by tactic by technique embedding with sub-techniques excluded

CVE	Vulnerability	Tactic ID	Tactic Title	Technique ID	Technique	Confidence
CVE-2021-26084	Atlassian Confluence Server and Data Center Object-Graph Navigation Language (OGNL) Injection Vulnerability	TA0008	Lateral Movement	T1072	Software Deployment Tools	0.597603526
		TA0007	Discovery	T1046	Network Service Discovery	0.580131992
CVE-2021-26855	Microsoft Exchange Server Remote Code Execution Vulnerability	TA0002	Execution	T1059	Command and Scripting Interpreter	0.633783337
		TA0002	Execution	T1203	Exploitation for Client Execution	0.633760294
CVE-2021-34527	Microsoft Windows Print Spooler Remote Code Execution Vulnerability	TA0004	Privilege Escalation	T1547	Boot or Logon Autostart Execution	0.64199945
		TA0002	Execution	T1053	Scheduled Task/Job	0.6371305
CVE-2021-40444	Microsoft MSHTML Remote Code Execution Vulnerability	TA0005	Defense Evasion	T1220	XSL Script Processing	0.715400308
		TA0002	Execution	T1203	Exploitation for Client Execution	0.703556555
CVE-2021-44832	Apache Log4j2 Remote Code Execution vulnerability	TA0002	Execution	T1059	Command and Scripting Interpreter	0.60849264
		TA0002	Execution	T1648	Serverless Execution	0.594095278
CVE-2022-22965	Spring Framework JDK 9+ Remote Code Execution Vulnerability	TA0002	Execution	T1203	Exploitation for Client Execution	0.614510523
		TA0002	Execution	T1648	Serverless Execution	0.600589988
CVE-2023-35315	Windows Layer-2 Bridge Network Driver Remote Code Execution Vulnerability	TA0008	Lateral Movement	T1210	Exploitation of Remote Services	0.611468556
		TA0002	Execution	T1059	Command and Scripting Interpreter	0.60796344

CVE mapping based on Gemini ATT&CK by tactic by technique embedding with sub-techniques excluded

As a means to vet the accuracy of above results, NopSec manually conducted the same exercise and mapped each CVE to two (2) ATT&CK TTPs, leveraging only the NVD-provided CVE description to evaluate the vulnerability. The table below captures the results of the manual analysis as well as the BERT and Gemini output, together with the two models' accuracy percentage compared to the manual mapping.

CVE	Vulnerability	Manual Mapping	BERT Mapping	Gemini Mapping	BERT vs Manual	Gemini vs Manual
CVE-2021-26084	Atlassian Confluence Server and Data Center Object-Graph Navigation Language (OGNL) Injection Vulnerability	Exploit Public-Facing Application (T1190)	Exploitation for Client Execution (T1203)	Software Deployment Tools (T1072)	50%	0%
		Data Manipulation (T1565)	Data Manipulation (T1565)	Network Service Discovery (T1046)		
CVE-2021-26855	Microsoft Exchange Server Remote Code Execution Vulnerability	Command and Scripting Interpreter (T1059)	Exploitation for Client Execution (T1203)	Command and Scripting Interpreter (T1059)	0%	50%
		Exploitation of Remote Services (T1210)	Exploitation for Defense Evasion (T1211)	Exploitation for Client Execution (T1203)		
CVE-2021-34527	Microsoft Windows Print Spooler Remote Code Execution Vulnerability	Exploitation for Privilege Escalation (T1068)	Event Triggered Execution (T1546)	Boot or Logon Autostart Execution (T1547)	0%	0%
		Exploitation of Remote Services (T1210)	Event Triggered Execution (T1546)	Scheduled Task/Job (T1053)		
CVE-2021-40444	Microsoft MSHTML Remote Code Execution Vulnerability	User Execution (T1204)	Exploitation for Client Execution (T1203)	XSL Script Processing (T1220)	50%	50%
		Exploitation of Remote Services (T1210)	Exploitation for Defense Evasion (T1211)	Exploitation for Client Execution (T1203)		
CVE-2021-44832	Apache Log4j2 Remote Code Execution vulnerability	Exploitation of Remote Services (T1210)	Steal or Forge Kerberos Tickets (T1558)	Command and Scripting Interpreter (T1059)	0%	0%
		Exploit Public-Facing Application (T1190)	Exploitation for Credential Access (T1212)	Serverless Execution (T1648)		
CVE-2022-22965	Spring Framework JDK 9+ Remote Code Execution Vulnerability	Exploit Public-Facing Application (T1190)	Exploitation for Client Execution (T1203)	Exploitation for Client Execution (T1203)	50%	50%
		Server Software Component (T1505)	Exploitation for Privilege Escalation (T1068)	Serverless Execution (T1648)		
CVE-2023-35315	Windows Layer-2 Bridge Network Driver Remote Code Execution Vulnerability	Device Driver Discovery (T1652)	Network Boundary Bridging (T1599)	Exploitation of Remote Services (T1210)	0%	0%
		Exploitation of Remote Services (T1210)	Windows Management Instrumentation (T1047)	Command and Scripting Interpreter (T1059)		
Legend:				**Overall Accuracy**	14.3%	21.4%
		TTP Exact match				
		TTP Plausibly related				
		TTP Unrelated				

Manually mapped CVEs compared against Gemini and BERT ATT&CK by tactic by technique embedding

NopSec found that Google Gemini was most closely aligned with manually mapped TTPs when mapping a CVE to a related ATT&CK technique based on the NVD description of a vulnerability, tactic description, and technique description. The results indicated that the high level tactic description failed to provide sufficient details to establish a reliable relationship between a CVE and an ATT&CK technique. This result was surprising. When NopSec approached this problem we assumed that by first filtering the results to a subset of related ATT&CK tactics overall accuracy would increase, however the opposite result was observed.

To better understand this result, we analyzed the textual descriptions of an ATT&CK Tactic and compared them to the ATT&CK Technique descriptions. In general, the descriptions of an ATT&CK tactic were concise and contained only high level information. Due to the limited semantic details present in the tactic description, it's not possible to establish a meaningful relationship. This method of mapping CVEs to ATT&CK techniques relied on filtering based on tactics first. Due to this behavior the resulting TTPs output by the LLMs using this technique did not accurately map to the vulnerability described by the CVE.

APPENDIX B

Mapping CVE by ATT&CK Technique with a Single Sentence

We encountered an interesting result when analyzing the mapping of CVEs to ATT&CK TTPs when comparing the similarity between CVE descriptions and the ATT&CK technique description. In the case of CVE-2021-26855, BERT, Gemini, and a human analyst all arrived at the same TTPs. This CVE was unique because the CVE description consisted of a single sentence. Based on this outlier result, NopSec limited the CVE input data to the first sentence of the CVE description when calculating similarity to TTPs, as captured in the results table below.

CVE	Vulnerability	Manual Mapping	BERT Mapping	Gemini Mapping	BERT vs Manual	Gemini vs Manual
CVE-2021-26084	Atlassian Confluence Server and Data Center Object-Graph Navigation Language (OGNL) Injection Vulnerability	Exploit Public-Facing Application (T1190)	Template Injection (T1221)	Template Injection (T1221)	50%	0%
		Data Manipulation (T1565)	Exploit Public-Facing Application (T1190)	Software Deployment Tools (T1072)		
CVE-2021-26855	Microsoft Exchange Server Remote Code Execution Vulnerability	Command and Scripting Interpreter (T1059)	Exploitation for Client Execution (T1203)	XSL Script Processing (T1220)	0%	0%
		Exploitation of Remote Services (T1210)	Exploitation for Defense Evasion (T1211)	Exploitation for Credential Access (T1212)		
CVE-2021-34527	Microsoft Windows Print Spooler Remote Code Execution Vulnerability	Exploitation for Privilege Escalation (T1068)	Exploitation for Privilege Escalation (T1068)	Exploitation of Remote Services (T1210)	50%	0%
		Exploitation of Remote Services (T1210)	Indirect Command Execution (T1202)	System Script Proxy Execution (T1216)		

CVE-2021-40444	Microsoft MSHTML Remote Code Execution Vulnerability	User Execution (T1204)	Exploitation for Client Execution (T1203)	XSL Script Processing (T1220)	50%	0%
		Exploitation of Remote Services (T1210)	Trusted Developer Utilities Proxy Execution (T1127)	Exploitation of Remote Services (T1210)		
CVE-2021-44832	Apache Log4j2 Remote Code Execution vulnerability	Exploitation of Remote Services (T1210)	Log Enumeration (T1654)	Log Enumeration (T1654)	0%	0%
		Exploit Public-Facing Application (T1190)	Video Capture (T1125)	Cloud Storage Object Discovery (T1619)		
CVE-2022-22965	Spring Framework JDK 9+ Remote Code Execution Vulnerability	Exploit Public-Facing Application (T1190)	Template Injection (T1221)	Exploit Public-Facing Application (T1190)	50%	50%
		Server Software Component (T1505)	Server Software Component (T1505)	Template Injection (T1221)		
CVE-2023-35315	Windows Layer-2 Bridge Network Driver Remote Code Execution Vulnerability	Device Driver Discovery (T1652)	Network Boundary Bridging (T1599)	Device Driver Discovery (T1652)	0%	50%
		Exploitation of Remote Services (T1210)	Hardware Additions (T1200)	Exploitation of Remote Services (T1210)		
Legend:				**Overall Accuracy**	28.6%	14.3%
		TTP Exact match				
		TTP Plausibly related				
		TTP Unrelated				

FIrst sentence of a CVE description compared against Gemini and BERT ATT&CK technique embedding

An analysis of the results didn't provide much additional insight or improved accuracy. It appears that BERT is more accurate when analyzing a limited amount of input text, which is in contrast to the Gemini framework. However, neither BERT nor Gemini returned a substantial number of relevant ATT&CK TTPs when input data was limited to a single sentence of a CVE vulnerability description.


NOPSEC
www.nopsec.com