PHOENIX SECURITY

A THREAT CENTRIC APPROACH ON VULNERABILITIES LEVERAGING LLM AND PREDICTIONS

Forging the path ahead together for better prioritization.

Through a unique, data-driven approach, this research connects root causes like input validation failures, code execution vulnerabilities, and memory corruption issues to their role in ransomware attacks, enabling organizations to prioritize patching and remediation efforts effectively. The findings provide a powerful predictive framework based on exploit availability, early indicators, and real-world exploitation patterns, empowering defenders to stay one step ahead.

1st Edition

Francesco Cipollone
Researcher & Co-Founder Phoenix Security

PHOENIX SECURITY

ACT ON RISK

A threat centric approach for vulnerabilities Exploitation Prediction

Evolving Vulnerability Exploit Prediction: Leveraging LLMs to move beyond EPSS and scaling CVE Problems leveraging threat patterns

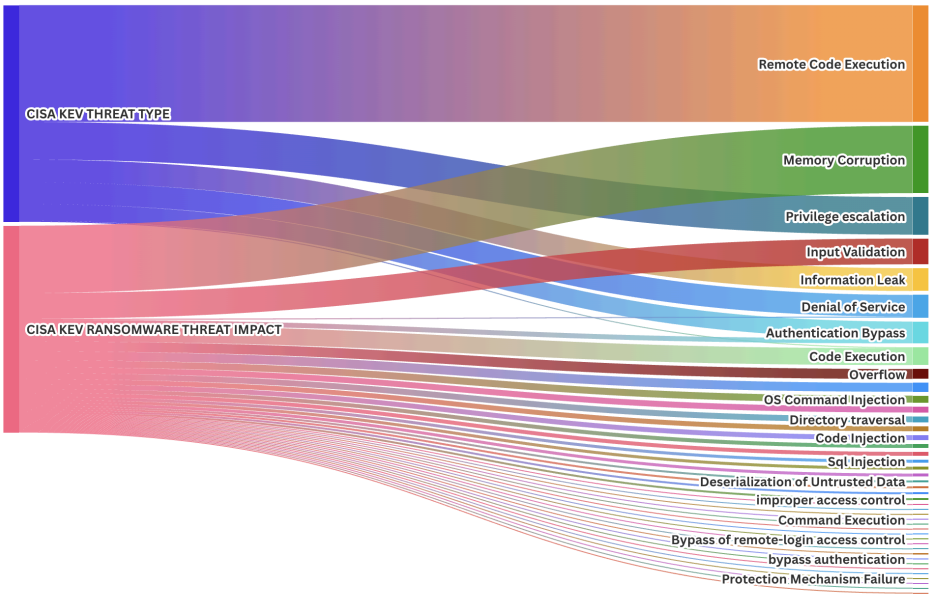
To all the vulnerability management professionals and researchers out there, thank you for all the hard work, this is a work of passion and in progress and hope to inspire the next generation of predictive analysis on vulnerabilities. Because burnout is something we can all fight together reducing the backlog of vulnerabilities and focusing on the most important work

Francesco Cipollone Researcher, Passionate advocate, CEO & Co-Founder Phoenix Security

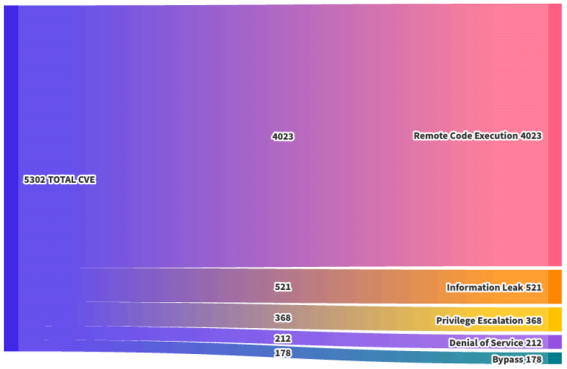
Index

Executive Summary

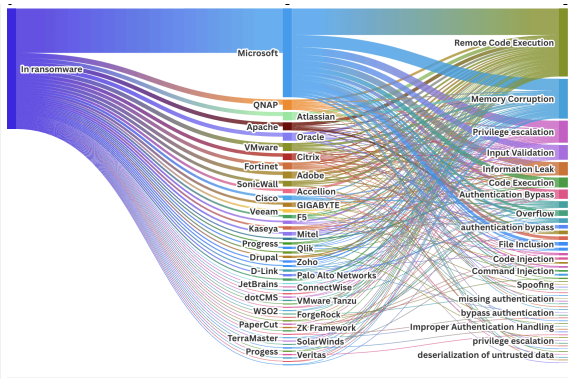
Purpose: This white paper investigates how specific vulnerability characteristics – particularly root cause and technical impact – can predict the likelihood of high-risk exploitation. It also demonstrates how LLMs can aid in the categorization effort at scale. By examining empirical data from Zero day analysis, Bug Bounty, Exploits used in ransomware, Vulncheck and *CISA's Known Exploited Vulnerabilities (KEV)* catalog and industry analyses, we illustrate that certain types of flaws (e.g. memory corruption or input validation failures leading to Remote Code Execution) are disproportionately leveraged in **ransomware campaigns** and zero-day attacks. The paper concludes by presenting a predictive framework that cybersecurity professionals can use to identify "likely-to-be-exploited" vulnerabilities before threat actors weaponize them.



CISA Exploits Used in Ransomware Threat Type and Threat Impact



Verified Exploits Threat Type



CISA Kev Threat Impact & in ransomware

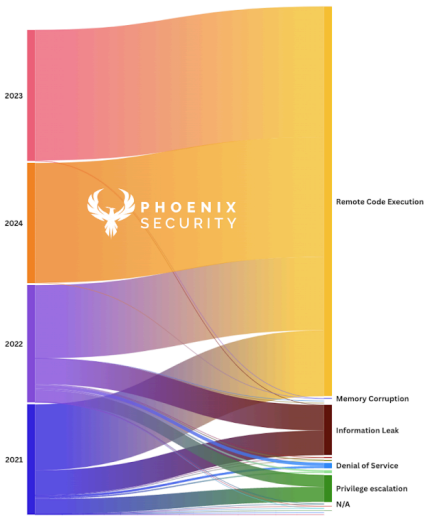
Key Findings: Vulnerabilities enabling **Remote Code Execution (RCE)** or **Privilege Escalation** dominate real-world exploits. For instance, RCE appears 133 times in the KEV catalog (historically the largest category)【15†L198-L206】. In contrast, lower-impact issues (like simple Denial-of-Service) rarely appear in active exploitation feeds【15†L210-L218】. Moreover, root causes such as **improper input validation** (e.g. SQL/command injection,

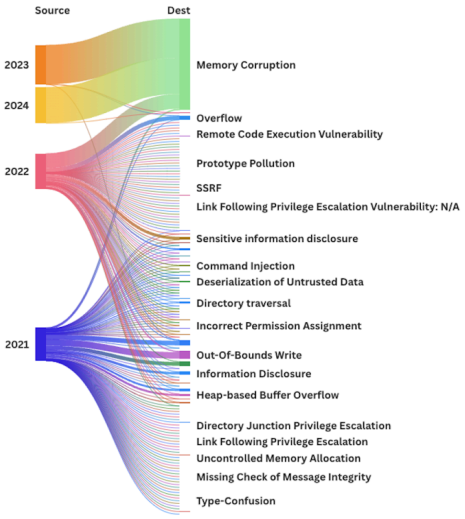
path traversal) and **memory corruption** (e.g. buffer overflows) are common denominators used by ransomware groups. A majority of *"ransomware-related"* CVEs fall into these categories, underscoring how flaws that allow arbitrary code execution are a magnet for attackers. Notably, out of roughly 1,225 KEV-listed vulnerabilities, **214** are explicitly tied to known ransomware campaigns【12†L231-L239】– a selective subset that overwhelmingly involves input validation failures or memory safety bugs.

Technical Impact methods in the NVD

Exploitation method in the NVD

Zero-Day Trend: The data also reveals a surge in zero-day exploitations. In 2023, Google's Threat Analysis Group tracked **97 zero-day vulnerabilities exploited in the wild**, over 50% more than the previous year (though slightly fewer than the 106 seen in 2021)【35†L301-L309】. Critical advisories from the "Five Eyes" security agencies confirm this shift: **11 of the top 15** routinely exploited CVEs in 2023 were initially exploited as *zero-day flaws*, compared to only 2 of 15 in the prior year【18†L93-L100】. This highlights that threat actors (including ransomware operators) are increasingly targeting fresh vulnerabilities *before patches are widely applied*. Crucially, those zero-days tend to share the same high-risk traits – allowing system takeover via code execution or authentication bypass.

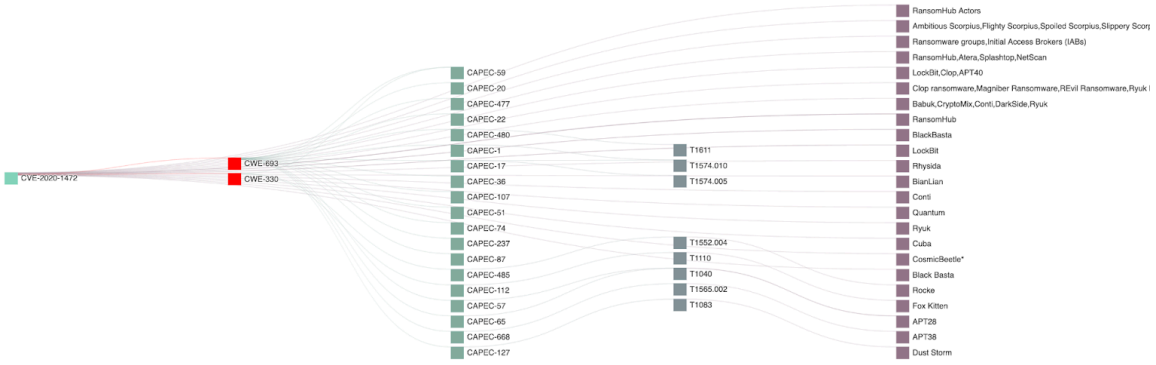




Zero Day Threats profile and impact analysis

Early Indicators: Another major finding is the importance of *early warning signals* like publicly available exploits and exploit proof-of-concepts (PoCs). The presence of a working exploit significantly raises the probability of real-world attacks【38†L227-L235】. Many headline vulnerabilities (e.g. Log4Shell, ProxyShell) saw functional exploits published within days of disclosure, followed by active use in ransomware incidents. Empirical scoring systems such as **EPSS** (Exploit Prediction Scoring System) quantify this risk: vulnerabilities with **exploit code released** or demonstrated tend to score high on likelihood and indeed correlate with subsequent ransomware targeting【9†L19-L22】. For example, within hours of a PoC release, the notorious Ryuk ransomware gang leveraged the *Zerologon* privilege-escalation bug (CVE-2020-1472) to blitz enterprise domain controllers【44†L512-L519】. Early indicators like this can forecast which vulnerabilities will become the "next big" exploits.

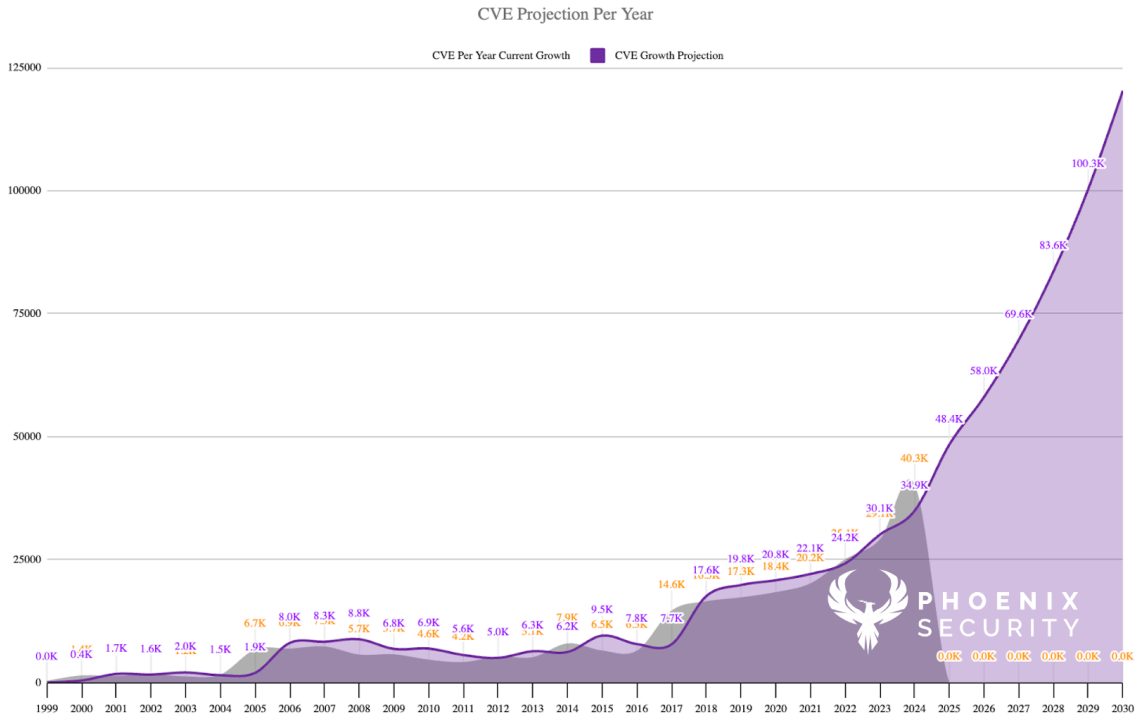
Strategic Implications: By mapping vulnerabilities' technical impact and root cause to known threat patterns, defenders can **forecast exploitation** and prioritize patches intelligently. The analysis shows that if a new CVE involves **memory corruption or input validation weaknesses enabling RCE**, and especially if a PoC exploit emerges, it should be treated as an imminent threat. Aligning remediation with these predictors can preempt ransomware attacks and limit zero-day exposure. In practice, this means merging vulnerability intelligence (CWE categories, exploit availability, KEV data) into risk scoring. In fact, industry guidance now urges vendors to tag CVEs with accurate CWE weakness codes to facilitate such root-cause-based analysis【18†L111-L115】. Overall, a threat-centric vulnerability management approach – focusing on *"Which flaws are most likely to be weaponized/used in ransomware?"* – can significantly improve an organization's resilience against ransomware and other high-impact cyber attacks.



Phoenix Security Intelligence combined with threat centric method for a threat based https://**ai-threat.phoenix.security**

Introduction

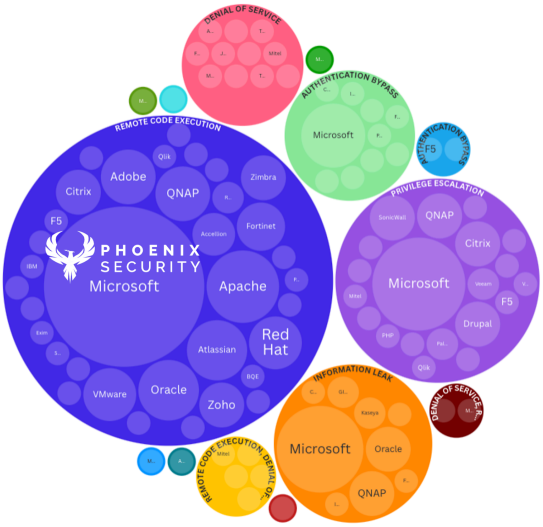
After Covid, many organizations have shifted to the cloud and started their digital transformation journey. The exploitation of vulnerabilities is lower than the current declaration, yet only a small fraction will ever be exploited in the wild【36†L47-L55】. The challenge for defenders is identifying *which* vulnerabilities are likely to become real threats (via ransomware, nation-state operations, or criminal exploitation) and prioritizing those for rapid mitigation. Traditional severity metrics alone (e.g. CVSS scores) are insufficient for this task – many *"critical"* CVEs never see active exploitation, while some lower-severity flaws are weaponized in major breaches. This disconnect has prompted a shift toward **threat-informed vulnerability management**, where characteristics like a vulnerability's root cause, impact, and exploit evidence are used to gauge the **likelihood of exploitation**.



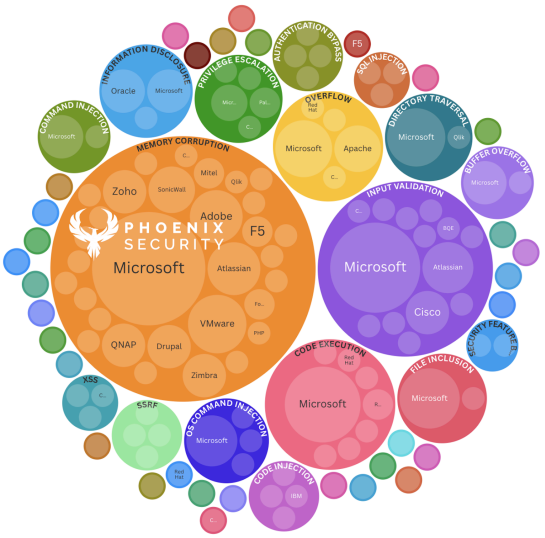
Vulnerability exploitation over the years (gray)with accelerated prediction based on current predicted acceleration (purple)

One of the clearest high-impact threat scenarios is **ransomware**. Ransomware operators aggressively exploit vulnerabilities that grant them quick remote access or elevated privileges, as these flaws streamline the path to deploying their payloads. Over the past few years, ransomware groups have increasingly behaved like advanced persistent threats, even burning zero-day exploits to infiltrate targets. For example, officials reported that malicious actors are *"increasingly exploiting zero day vulnerabilities to compromise enterprise networks"*【18†L86-L94】. The stakes are high: a single unpatched high-risk vulnerability can lead to domain-wide ransomware deployment or data extortion, causing . Indeed, an annual joint advisory on top exploited flaws revealed that **zero-days comprised the majority of routinely exploited vulnerabilities in 2023**, a sharp rise from 2022【17†L66-L74】【

18†L93-L100】. Attackers have grown adept at rapidly operationalizing new exploits, leaving ever-smaller windows for defenders.



Root Cause analysis of vulnerabilities used in Ransomware

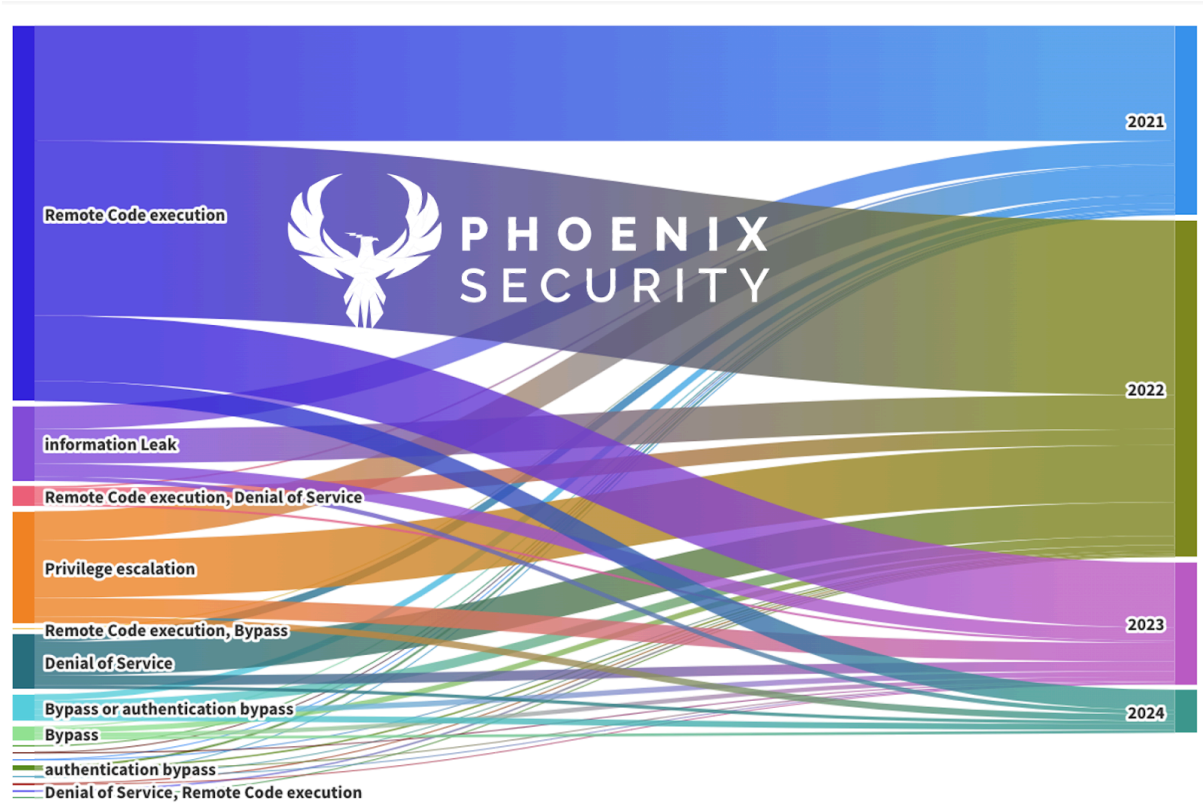


Impact analysis of vulnerabilities used in Ransomware

Research Objective: This paper posits that *vulnerability attributes themselves* – specifically the underlying **weakness (root cause) and technical impact – can serve as reliable predictors of exploitation risk. The paper demonstrates how to leverage LLMs to classify and re-classify vulnerabilities in order to fix the 40% gap in vulnerability threat type that the NVD currently has*. By using vulnerability datasets and real incidents, we aim to show patterns that link certain CWE categories and impact types to eventual ransomware attacks and zero-day use. For instance, is an **improper input validation** flaw (like an injection vulnerability) more likely to be picked up by ransomware gangs? Do **memory corruption** bugs in popular software have a higher chance of being exploited as zero-days? And how do early indicators like **exploit availability** or a "**verified exploit**" flag factor into forecasting attacks?

The paper begins by analyzing vulnerability datasets before describing how we leveraged LLMs to extract additional information regarding attack vectors, likelihood, and impact. We then apply the enhanced dataset to a predictive formula to determine exploit likelihood. We conclude with some examples where the predictive analysis powered by the enhanced data led to real world results.

To answer these questions, we draw on multiple data sources:



Analysis of all the vulnerabilities across the years

- **CISA Known Exploited Vulnerabilities (KEV) Catalog:** A curated list of CVEs known to be exploited in the wild, maintained by the U.S. Cybersecurity and Infrastructure Security Agency. This offers a ground truth of which vulnerabilities *actually* have been used by attackers【15†L179-L187】. Within KEV, a subset is marked as *"Known to be used in ransomware campaigns"* – these will be examined to extract common traits.
- **Phoenix Security Vulnerability Data:** Deep crawling and evidence based data from Ransomware, Exploit and analysis from Phoenix Security's threat intelligence platform, which cross-references KEV data with vulnerability attributes (CWE weakness types, impact categories, etc.). This provides statistical insight, such as the distribution of vulnerability types in KEV over recent years【15†L198-L206】 and specialized views like the frequency of certain CWEs among ransomware-exploited CVEs【37†L307-L315】.
- **Zero-Day Exploitation Reports:** Notably Google's Project Zero and Threat Analysis Group report on 2023 in-the-wild zero-days【35†L301-L309】, as well as joint cybersecurity advisories. These highlight trends in how many zero-days are being exploited yearly and in what contexts (e.g. a rise in exploits against network devices and VPNs, which are commonly targeted by ransomware affiliates).
- **Exploit Availability Data:** Information on whether public exploits or proofs-of-concept exist for vulnerabilities (e.g. from Exploit-DB, Metasploit, or VulnCheck data). Prior research like EPSS has quantified that the **presence of a publicly available exploit** is a strong predictor of exploitation likelihood【

38†L227-L235】. We incorporate this dimension to emphasize the role of early exploit release as an “advance warning” of attacks.

Using these inputs, we perform a correlation analysis between **vulnerability characteristics** (such as CWE category or impact type) and **real-world threat outcomes** (such as inclusion in ransomware campaigns or appearance in KEV/zero-day lists). The intention is to build a data-driven case that you can *forecast* which newly disclosed vulnerabilities are most likely to become high-risk **ransomware entry points or zero-day exploits**, based on their intrinsic features. Ultimately, this will feed into a **predictive framework** for vulnerability risk management – a model to help security teams get ahead of attackers by patching or monitoring the vulnerabilities that truly matter, before they are weaponized.

Applying and Integrating LLM-Based Classification for Enhanced Threat Analysis

An effective application of large language models (LLMs) to vulnerability classification follows a staged process that ensures the final system accurately identifies and prioritizes high-risk weaknesses. Our creation of a threat-centric LLM agent for vulnerability classification stemmed from a methodical, three-phase process designed to yield high-fidelity risk assessments that align with real attacker behavior:

1. Building the Authoritative Dataset

We began by expanding an initial sample of 50 pre-labeled vulnerabilities to a robust dataset of 500. This growth used data from recognized sources (KEV, NVD) and employed LLM-based insights to correct or augment existing labels. Each entry in the resulting dataset incorporates granular root-cause tags (e.g., memory corruption, injection), contextual impact (e.g., RCE vs. information disclosure), and threat intelligence markers (e.g., verified exploits). Relevant metadata—like CVSS base score, vendor advisories, exploit maturity—enables the model to learn nuanced correlations between an issue’s technical descriptors and its actual exploitation in the wild. To prevent spurious or speculative reasoning (“hallucinations”), we constrained prompt structures and strictly validated data against known facts.

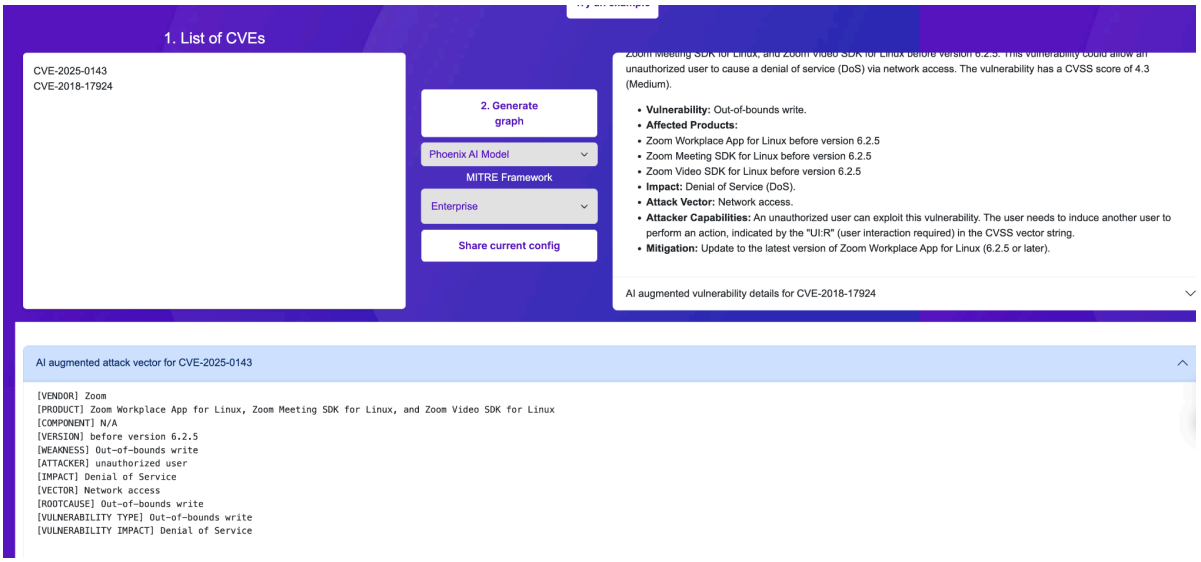
2. Enhancing with Retrieval-Augmented Generation (RAG)

After the initial fine-tuning, we integrated retrieval-augmented generation instructions, letting the model query external knowledge bases. These repositories hold structured and unstructured information: newly discovered PoCs, updated EPSS feeds, or validated exploit analysis from CTI sources. By drawing on these up-to-date materials during inference, the LLM adapts classifications to reflect emerging techniques or attacks. This dynamic layer keeps the agent accurate long after initial training, bridging static data with continuous updates from Phoenix Security’s threat intelligence, among others.

3. Deploying the Threat-Centric Agent

In the final stage, we deployed the LLM as an automated agent that scales vulnerability categorization across the organization. Upon encountering a new disclosure or zero-day, it cross-references historically weaponized categories (e.g., known injection vectors used by ransomware), exploit likelihood (via EPSS or other data feeds), and environment specifics (e.g., critical system or publicly exposed). The agent synthesizes these factors into a risk-oriented verdict—flagging high-priority issues for immediate patching while offering deeper context on root cause and parallel exploit patterns. By uniting real-time data ingestion with robust classification rules, Phoenix's threat-centric agent ensures defenders focus resources on the vulnerabilities adversaries are most likely to exploit.

You can see a preview of the agent at https://**ai-threat.phoenix.security**



Understanding Better the vulnerability data and impact leveraging LLM

Whilst the description of vulnerabilities are great they don't really give a lot of context why a vulnerability is bad if we take the example of CVE-2023-2868 it clearly shows. You will see that the LLM augmented research information goes into much greater detail than the CVE description.

The *Barracuda Email Security Gateway* vulnerability (CVE-2023-2868), which was an input parsing flaw that allowed remote code execution by sending a crafted email attachment. It's explicitly listed as **Improper Input Validation** in type【31†L264-L271】and was widely exploited (initially by state actors and later others). This again underscores that if user input (even something like an email file) isn't handled safely and leads to code injection, attackers will seize it.

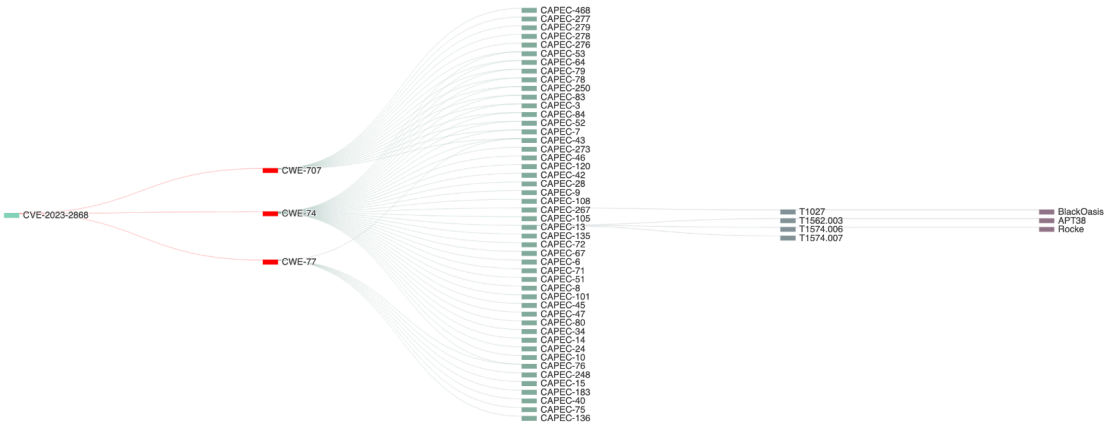
NVD CVE-2023-2868

By selecting these links, you will be leaving the NIST webspace.

A remote command injection vulnerability exists in the Barracuda Email Security Gateway (appliance form factor only) product effecting versions 5.1.3.001-9.2.0.006. The vulnerability arises out of a failure to comprehensively sanitize the processing of .tar file (tape archives). The vulnerability stems from incomplete input validation of a user-supplied .tar file as it pertains to the names of the files contained within the archive. As a consequence, a remote attacker can specifically format these file names in a particular manner that will result in remotely executing a system command through Perl's qx operator with the privileges of the Email Security Gateway product. This issue was fixed as part of BNSF-36456 patch. This patch was automatically applied to all customer appliances.

CWE-ID	CWE Name	Source
CWE-77	Improper Neutralization of Special Elements used in a Command ('Command Injection')	NIST
CWE-20	Improper Input Validation (augmented later)	Google Inc.

AI Augmented Vulnerability analysis of CVE-2023-2868



https://ai-threat.phoenix.security analysis

AI augmented attack vector for CVE-2023-2868
[VENDOR] Barracuda
[PRODUCT] Email Security Gateway (ESG)
[COMPONENT] Filename handling in .tar file processing
[VERSION] 5.1.3.001-9.2.0.006
[WEAKNESS] Incomplete Input Validation

[ATTACKER] Remote attacker
[IMPACT] Remote command execution
[VECTOR] Sending a specially crafted email with a malicious TAR file attachment
[ROOTCAUSE] Incomplete input validation of user-supplied filenames within `.tar` files
[VULNERABILITY TYPE] Command Injection
[VULNERABILITY IMPACT] Code execution

Full AI Augmented Vulnerability analysis of CVE-2023-2868

The provided documents contain detailed information regarding CVE-2023-2868, a remote command injection vulnerability in the Barracuda Email Security Gateway (ESG) appliance.

Here's a breakdown of the vulnerability, its exploitation, and related information:

Root Cause:

- Incomplete Input Validation: The vulnerability stems from incomplete input validation of user-supplied `.tar` files, specifically the filenames contained within the archive.
- Command Injection: A remote attacker can format filenames in a specific manner that results in remotely executing a system command through Perl's `qx` operator with the privileges of the Email Security Gateway product.
- Vulnerable Code: The specific code identified is `qx{$tarexec -O -xf $tempdir/parts/$part '$f'};`, where `$f` is a user-controlled variable containing filenames from the TAR archive, which is then executed as a system command without proper sanitization.

Vulnerability Details:

- Affected Product: Barracuda Email Security Gateway (ESG) appliance (not SaaS versions).
- Affected Versions: 5.1.3.001-9.2.0.006.
- Attack Vector: Sending a specially crafted email with a malicious TAR file attachment to a vulnerable ESG appliance.

Impact of Exploitation:

- Initial Access: Remote command execution, leading to initial access on the ESG appliance.
- Reverse Shell: Execution of reverse shell payloads, providing the attacker with command-line access.
- Backdoor Installation: Downloading and installing secondary backdoor payloads (e.g., SEASPY, SALTWATER, SEASIDE) for persistent access.
- Data Exfiltration: Staging and exfiltration of email data and SSL certificates.
- Lateral Movement: Reconnaissance activity within the victim network using tools like `fscan`.

Attack Vector and Attacker Capabilities:

- Attacker Skill: The attacker is described as "aggressive and highly skilled."
- Email Delivery: The attacker sent emails with malicious TAR attachments. The emails often had generic subjects and bodies to appear as spam. Spoofed email addresses and compromised ESG appliances were used as sending points.
- Infrastructure: The attacker used VPS servers (e.g., Vultr), IP addresses allocated to China Telecom, and infrastructure overlapping with other suspected China-nexus espionage operations.

Actor Details:

- Attribution: Mandiant assesses with high confidence that the threat actor, tracked as UNC4841, conducted espionage activity in support of the People's Republic of China (PRC).
- Targeting: Targeted organizations spanned public and private sectors worldwide, with a focus on the Americas. A significant portion of victims were government agencies. Focused data exfiltration targeted email domains and users from ASEAN Ministry of Foreign Affairs (MFAs), foreign trade offices, and academic research organizations in Taiwan and Hong Kong.

Malware Families:

- SALTWATER: A trojanized module for the Barracuda SMTP daemon (bsmtpd) providing backdoor functionality, file upload/download, command execution, proxy, and tunneling capabilities.
- SEASPY: An x64 ELF persistence backdoor that poses as a legitimate Barracuda Networks service and establishes itself as a PCAP filter monitoring SMTP traffic. It's activated by a "magic packet" and has code overlap with the publicly available backdoor `cd00r`.
- SEASIDE: A Lua-based module for the Barracuda SMTP daemon (bsmtpd) that monitors SMTP HELO/EHLO commands to receive a C2 IP address and port, which are then used to establish a reverse shell.
- SANDBAR: A rootkit in the form of a trojanized network file system kernel module for linux (nfsd_stub.ko) that contains hooks to hide processes that begin with a specified name.
- SEASPRAY: A launcher written in Lua that is a trojanized Barracuda email security gateway module.
- SKIPJACK: A passive backdoor written in Lua that is a trojanized version of a Barracuda email security gateway module that processes emails.
- WHIRLPOOL: A C based utility used to create a TLS reverse shell.

Persistence Mechanisms:

- Cron jobs to execute reverse shells.
- Adding SEASPY execution to `/etc/init.d/rc` and the `update_version` Perl script.
- Deployment of the SANDBAR kernel rootkit.

Barracuda's Response and Recommendations:

- Patches: Security patches were applied to all ESG appliances worldwide to remediate the vulnerability.
- Containment Script: A script was deployed to impacted appliances to contain the incident and counter unauthorized access.
- Recommendation: Barracuda recommends immediate replacement of compromised ESG appliances, regardless of patch level. Impacted customers were notified via the ESG user interface and direct outreach.
- Credential Rotation: Customers are advised to rotate credentials connected to the ESG appliance (LDAP/AD, Barracuda Cloud Control, FTP, SMB, TLS certificates).
- Network Monitoring: Review network logs for IOCs and unknown IPs.
- Hardening Guide: Barracuda and Mandiant provided detailed hardening recommendations.

Indicators of Compromise (IOCs):

- A comprehensive list of network and endpoint IOCs, including IP addresses, domains, filenames, and hashes, is provided in the Mandiant blog post and Barracuda security updates.
- YARA and SNORT/Suricata rules are provided for detection.

Data and Methodology

As disclosure several elements of the paper refer to Phoenix Security data and methodology. Phoenix Security's analysis is powered by a comprehensive data collection and intelligence framework. We aggregate and analyze data from a variety of sources, including exploits observed in the wild, both commercial and non-commercial threat intelligence feeds, bug bounty data, zero-day vulnerability disclosures, and deep link analysis of security advisories and threat actor communications. This multi-faceted approach allows us to build a detailed understanding of the threat landscape. By correlating exploit patterns with vulnerability characteristics, we can identify emerging trends, understand threat actor methodologies, and ultimately predict which vulnerabilities are most likely to be weaponized. This rich dataset, combined with AI-driven analysis, forms the foundation for our threat-centric vulnerability management platform, https://ai-threat.phoenix.security .

Data Sources

- **CISA KEV Catalog:** We obtained the latest CISA Known Exploited Vulnerabilities list (which, as of late 2024, contains over 1,200 entries【12†L231-L239】). Each KEV entry includes the CVE, affected products, and a brief description. Importantly, KEV entries are vulnerabilities with **confirmed exploitation in the wild**, serving as a binary label of “exploited” for our analysis. The KEV catalog has an inherent bias towards widely used software and critical infrastructure vulnerabilities【15†L179-L187】, since it reflects those issues that CISA and partners have observed being actively leveraged by threat actors. Within the KEV data, we paid special attention to the **“ransomware” tag** introduced via CISA’s Ransomware Vulnerability Warning Pilot. CISA highlights certain KEV entries that are *“known to be used in ransomware campaigns”*, effectively a curated subset of KEV that ransomware actors have adopted. According to Phoenix Security’s analysis, out of the 1,225 KEV entries, **214** were earmarked by CISA as associated with ransomware exploitation【12†L231-L239】. This subset provides a focused lens on what vulnerability traits ransomware groups gravitate towards.
- **Phoenix Security Analysis & Enrichment:** Using Phoenix Security’s AI-based vulnerability intelligence tools and threat centric analysis tool https://**ai-threat.phoenix.security** , we enriched the KEV dataset with additional attributes:
- **Technical Impact Categories:** Phoenix’s platform categorizes vulnerabilities by their high-level impact (Remote Code Execution, Privilege Escalation, Denial of Service,

Information Disclosure, Authentication Bypass, etc.). This allowed us to compute frequencies of each impact type within the exploited sets. For example, Phoenix's published analysis shows how often RCE, PrivEsc, etc., have appeared in KEV each year【15†L198-L206】

- **CWE Weakness Types:** We leveraged mappings to Common Weakness Enumerations to identify the root cause category of each vulnerability. This includes whether a vulnerability is due to *"Improper Input Validation"*, *"Buffer Overflow (Out-of-Bounds Write/Read)"*, *"Missing Authentication"*, and so on. We cross-referenced these with the CWE Top 25 list to see which dan ss classes are most represented among exploited CVEs.
- **Threat Actor Tags:** In some cases, Phoenix data and external intel link specific CVEs to known threat actors or malware campaigns. This is valuable for case studies (e.g. tying CVE-2023-4966 to LockBit ransomware【41†L307-L310】, or CVE-2023-27350 to the Bl00dy ransomware gang【44†L493-L500】).
- **Zero-Day Reports:** We incorporated statistics from Google's **"Year in Review of 0-days exploited in 2023"**【35†L301-L309】 and the multi-agency **"Top Routinely Exploited Vulnerabilities" (2023)** advisory【18†L93-L100】 etc.) and specific examples of high-profile zero-day exploits. The alignment (or divergence) between KEV (which includes both zero-day and post-patch exploits) and zero-day trends can reveal patterns; for instance, whether certain vulnerability types are **consistently exploited immediately (as 0-day)** versus those typically exploited after disclosure.
- **Exploit Availability & EPSS:** We factored in data about exploit code availability. This included checking sources like Exploit-DB, Metasploit, and GitHub for PoCs corresponding to the vulnerabilities in our dataset, as well as referencing the Exploit Prediction Scoring System. *EPSS v3* (2023) explicitly uses features such as "public exploit exists" and software popularity to predict the 30-day exploit probability【38†L227-L235】. We reviewed the EPSS scores or rankings for certain CVEs where available, to see if high-EPSS vulnerabilities align with those exploited by ransomware (prior studies indicate they often do【9†L19-L22】). While we did not retrain any predictive model here, these data points support our qualitative correlation analysis.

Methodology: Our analysis proceeded in the following steps:

1. **Labeling and Categorization:** We labeled each vulnerability in the NVD, Zero Day, Ransomware and all the datasets with its **impact type** (e.g. RCE, Privilege Escalation, DoS, etc.) and its **root cause CWE category**. Many CVEs have multiple impacts or weaknesses; we focused on the primary impact for exploitation (e.g. if a bug allows RCE, that trumps secondary impacts). For root cause, we leveraged phoenix threat centric analysis and cross checked with existing mapped CWE , nonetheless due to the absence of reliable CWE mappings we relied on the former. Where KEV entries spanned multiple years, we noted the year of addition to see temporal trends. We proceeded with the same classification across datasets, from Zero day, to Exploits verified (github exploit verified) to ransomware
2. **Frequency Analysis:** We calculated how often each category appeared:
    - In **KEV overall** (to establish a baseline of what attackers exploit generally).
    - In the **ransomware-related subset** (to see which weaknesses and impacts are over-represented).

- In **zero-day exploits** reported in 2021–2023 (to see if the same categories appear in fresh exploits).

This yielded statistics like “X% of ransomware-related CVEs are RCEs” or “the count of memory corruption exploits in KEV for 2023 vs 2024”.

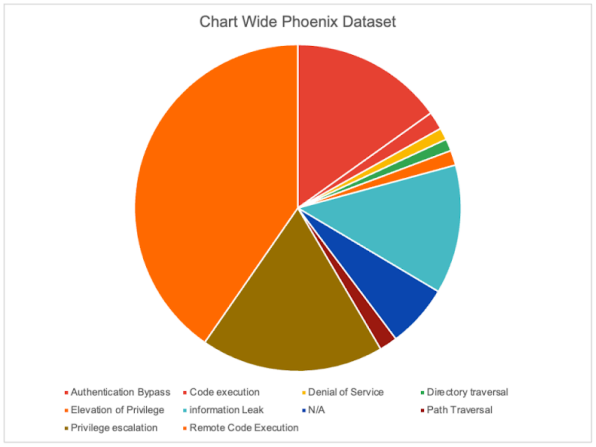
3. **Correlation and Cross-Tabulation:** We examined the overlap between different attributes. For example, we looked at how many of the ransomware-tagged CVEs had publicly known exploits versus those that did not at disclosure time. We also cross-tabulated CWE categories with impact types (often there’s a direct link: e.g. buffer overflow → RCE, or missing auth → PrivEsc). The aim was to identify *clusters* of high-risk characteristics – e.g. **“input validation flaw leading to code execution with exploit code available”** – that strongly correlates with exploitation by ransomware.
4. **Case Study Selection:** To ground the statistics in real-world context, we selected representative **case studies** of vulnerabilities that progressed from disclosure to exploitation to ransomware impact. We intentionally picked cases spanning different root causes and years:
   - An input validation/injection case (recent mass exploitation event).
   - A memory corruption RCE case (including a zero-day scenario).
   - An elevation of privilege case (used post-compromise in ransomware attacks).

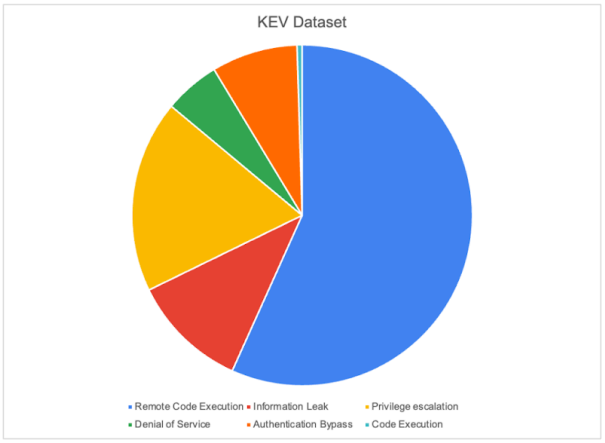
For each case, we gathered timeline information (when disclosed, when an exploit appeared, when used by which threat actor) from threat intelligence reports.

5. **Predictive Framework Synthesis:** Based on the above analysis, we formulated a framework of **predictive indicators**. We distilled the common factors present in the vulnerabilities that *did* become high-risk exploits. This framework was then validated against a few known exploited CVEs to see if it would have flagged them as likely threats. The framework’s components (like CWE type, exploit availability, etc.) are explained in the later section, with justification from the data (e.g. citing how often a given indicator was true for ransomware-exploited CVEs).

Note on Bias and Limitations: The KEV catalog skews towards certain vendors (e.g. Microsoft, Cisco, Adobe) and may under-represent application-layer issues (since it focuses on widely exploited ones). Also, not every vulnerability with a given trait will be exploited; our goal is to highlight higher *probabilities*, not certainties. We also recognize that attacker choices are dynamic – what was true in 2021 may evolve as defenses improve or new exploit techniques arise. However, the use of multi-year data and broad sources aims to capture persistent trends. All data analysis was done with these caveats in mind, and our conclusions favor *strategic insight* over hard prediction. We proceed now to present the core analysis of how vulnerability characteristics correlate with exploitation in ransomware and zero-day contexts.

This is particularly true analysing those two dataset on ransomware, where is evident that the KEV database is focused on specific type of vulnerabilities whilst the Phoenix security database has more widespread attacks





Dataset Comparison CISA KEV vs Wide ransomware limitations

KEV Dataset

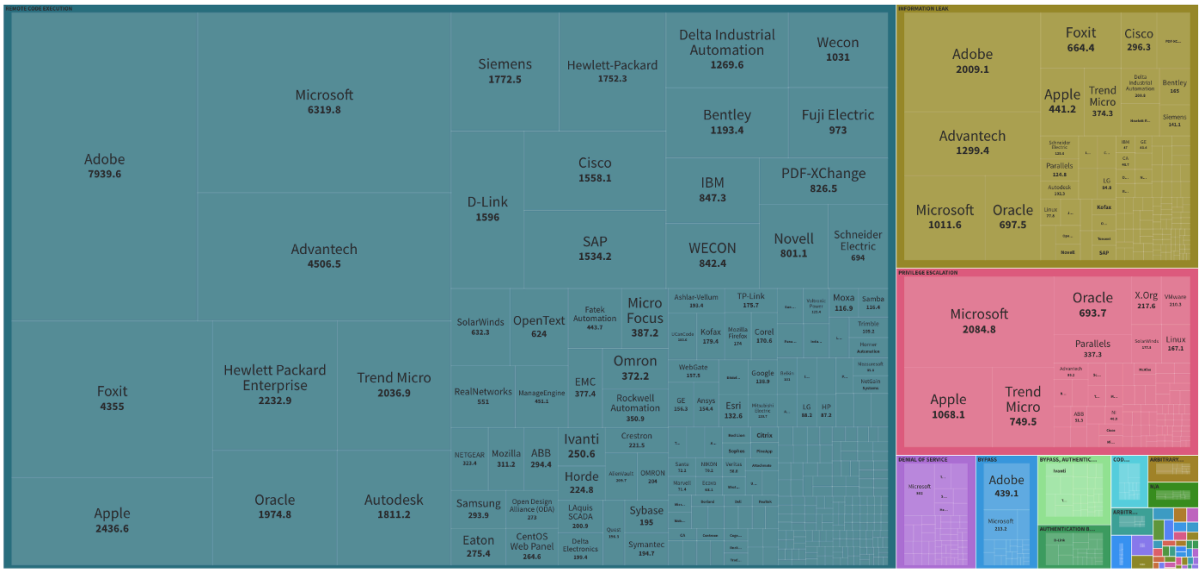
Remote Code Execution	57%
Information Leak	11%
Privilege escalation	18%
Denial of Service	5%
Authentication Bypass	8%
Code Execution	0%

Phoenix Security dataset / Non CISA KEV

Authentication Bypass	15.1%
Code execution	1.8%
Denial of Service	1.2%

Directory traversal	1.2%
Elevation of Privilege	1.5%
information Leak	12.8%
N/A	6.2%
Path Traversal	1.8%
Privilege escalation	18.1%
Remote Code Execution	40.4%

Core Analysis



Phoenix Security analysis of zero day vulnerabilities

Impact Type vs. Exploitation Likelihood

Remote Code Execution (RCE) as a Driver of Risk: Our analysis confirms a well-known intuition: vulnerabilities that allow an attacker to execute arbitrary code on the target system are the most coveted and frequently exploited. Within the CISA KEV dataset, **Remote Code**

Execution is the single most prevalent impact type among recorded exploits. Phoenix Security's automated categorization tallied RCE appearances at **133 instances** in KEV (across all years)【15†L198-L206】. This outnumbers other impact categories by a significant margin. RCE-oriented bugs are essentially giving attackers the keys to the kingdom, which explains why ransomware operators, APTs, and cybercriminals all prioritize them. Even in CISA's *2022 analysis* of top exploited vulnerabilities, RCE and code injection issues were front and center, and this trend continued into 2023【14†L31-L39】. Notably, RCE's dominance in KEV saw a slight decline in new entries by 2024 (only 8 RCEs added in early 2024 vs 52 in 2023)【15†L200-L208】, but this is attributed to yearly variability or possibly improved early patching, rather than attackers losing interest in RCE. Simply put, if a vulnerability can directly spawn a shell or run malware on a target, it has a high chance of exploitation.

Attack Type by threat actors and frequency of threat types used in ransomware

Other high-impact types show a similar story:

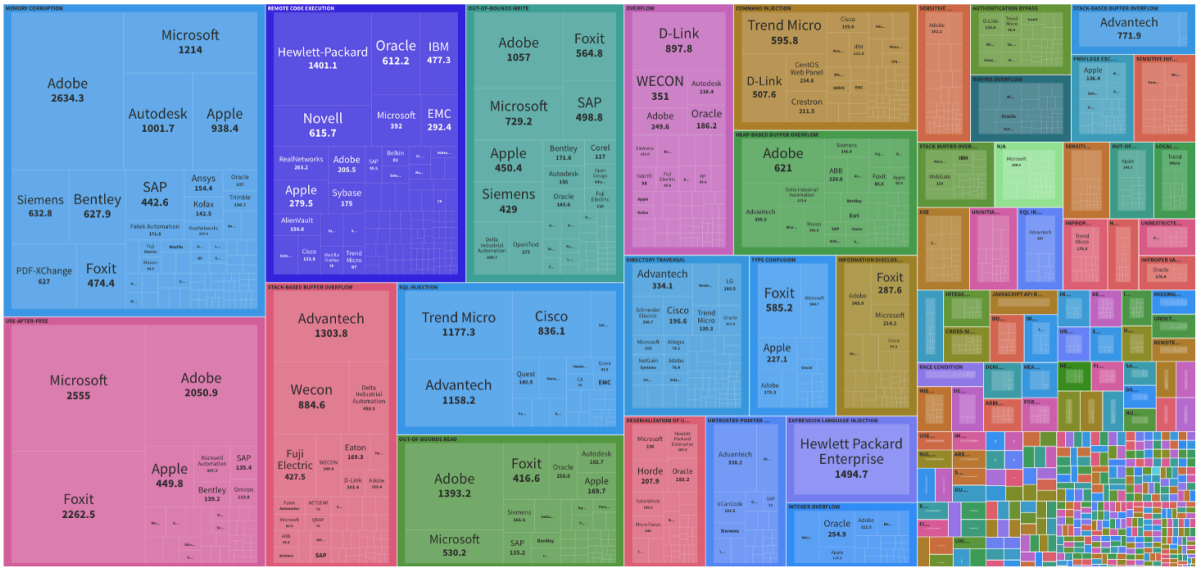
- **Privilege Escalation (Elevation of Privilege, EoP):** These flaws (often local vulnerabilities or post-initial-access issues) rank second in prevalence. In 2023, KEV entries for Privilege Escalation spiked (105 cases) and then dropped in 2024 (12 cases)【15†L201-L208】, reflecting perhaps the large batch of Microsoft Windows EoP vulnerabilities exploited by attackers (e.g. PrintNightmare and others) which have since tapered off. Privilege escalation vulnerabilities are critical in the ransomware kill-chain after an initial foothold is gained – for example, the infamous **Zerologon** bug (CVE-2020-1472) allowed threat actors to instantly promote themselves to domain admin after penetrating a network, and it remained one of the most exploited vulns even three years after disclosure【44†L503-L512】【44†L512-L519】. Our data shows PrivEsc vulnerabilities constitute a large portion of the KEV as well, confirming

that attackers reliably exploit *"privilege jumps"* to widen access. Ransomware groups have weaponized these to move laterally and deploy payloads enterprise-wide.

- **Denial of Service (DoS):** By contrast, pure DoS vulnerabilities (which merely crash services or exhaust resources) are rarely observed in the wild *unless* they can be leveraged for something more. The KEV catalog contains relatively few DoS-only CVEs (and those that do appear often were used to disrupt systems as a smoke screen or were combined with another exploit). The data shows DoS entries "remained relatively stable but low" in KEV【15†L210-L218】. This suggests that while DoS vulnerabilities can be severe in impact from an availability perspective, they are not a preferred tool for financially motivated actors like ransomware gangs (who seek access, not just service disruption). They may occasionally show up in state-sponsored attacks for destructive purposes, but they are not predictive of ransomware threat by themselves.
- **Information Disclosure:** Vulnerabilities that leak sensitive information (without direct code execution) have a moderate presence in KEV. In 2023 there were around 63 info-leak KEV entries, dropping to 8 in 2024【15†L201-L208】. While data exposure is a problem, these flaws usually need to be paired with another vulnerability to fully compromise a system (or else they aid reconnaissance). As such, info disclosure weaknesses (like an error that reveals passwords or a config file path) are often stepping stones. Ransomware actors typically prefer one-shot exploits that get them in, but they have used info leaks to expedite finding targets – for example, a path traversal that reveals user credentials which are then used to log in. Still, by themselves, info-leak vulns are less likely to predict ransomware incidents unless they facilitate a larger exploit chain.
- **Authentication Bypass/Impersonation:** This category (which includes missing or broken authentication, allowing unauthorized access) consistently appears in exploited sets but in lower numbers than RCE/EoP. KEV data showed authentication bypass vulnerabilities "consistent but relatively low in number"【15†L210-L218】, with a slight uptick in 2024. These can be potent – e.g., a hard-coded credential or logic flaw that lets an attacker in without credentials is essentially an open door. One example is CVE-2021-40539 (Zoho ManageEngine ADSelfService Plus bypass) which was exploited by advanced attackers. In ransomware context, authentication bypasses do feature: for instance, an **Authentication Bypass** in JetBrains TeamCity (CVE-2023-42793) allowed code execution and was actively exploited by state actors in 2023【44†L523-L532】【44†L533-L541】. However, such flaws are fewer compared to injection or buffer overflow issues that achieve similar results. They are extremely dangerous when they arise (since they often imply RCE or admin access without normal checks), so they form part of our high-risk criteria, albeit less commonly than explicit RCE bugs.

Insights: The overarching insight is that **vulnerabilities enabling an attacker to run commands or malware (RCE)** or **take control of accounts (Privilege Escalation or Auth Bypass)** have a dramatically higher likelihood of exploitation. The KEV catalog and threat reports reinforce this: *Remote Code Execution, Privilege Escalation, SQL Injection, and similar critical effects top the list of exploited vulnerabilities*【14†L31-L39】. Meanwhile, purely **integrity or availability impacting issues (like data tampering or DoS)**, without a direct path to code execution, are rarely weaponized in automation-driven attacks like ransomware. Thus, when predicting the next big threat, looking at the **impact type** is

paramount. If the impact is RCE or admin access, *assume the adversary is interested*. Impact alone isn't the full story, though – many vulnerabilities can lead to RCE, but through different underlying mechanisms. We next analyze how the **root cause or vulnerability type** (the CWE weakness) correlates with exploitation, especially in ransomware campaigns.

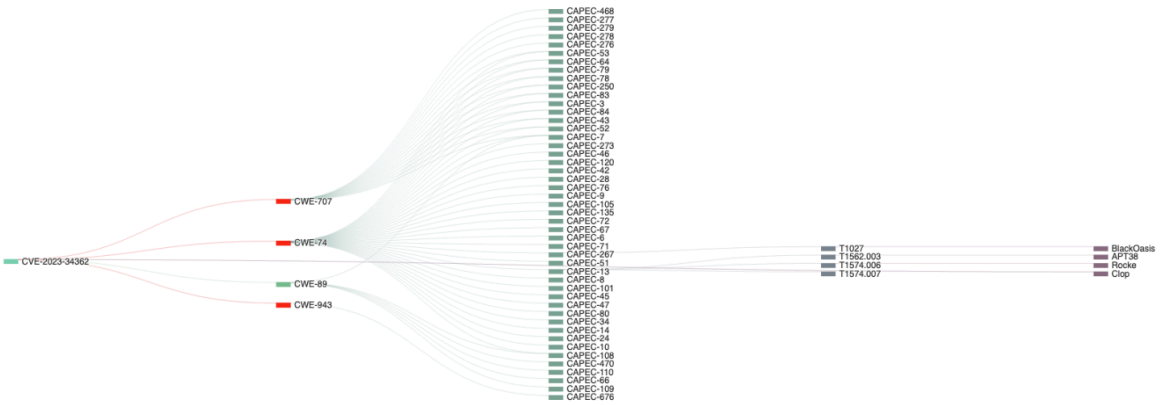


[Weaknesses in zero day by Phoenix security analysis]

Examples of Root Cause and CWE Patterns in Exploited Vulnerabilities

Impact tells us *what* a vulnerability allows an attacker to do (e.g. execute code), but the **root cause** tells us *how* the vulnerability works. Understanding the root cause is key to assessing how easily attackers can exploit it and how reliably it maps to their tactics. We examined the prevalent CWE weakness categories among exploited CVEs, with a focus on those linked to ransomware operations. A clear pattern emerged: **Improper input validation and memory safety issues are the predominant root causes** behind the most dangerous vulnerabilities.

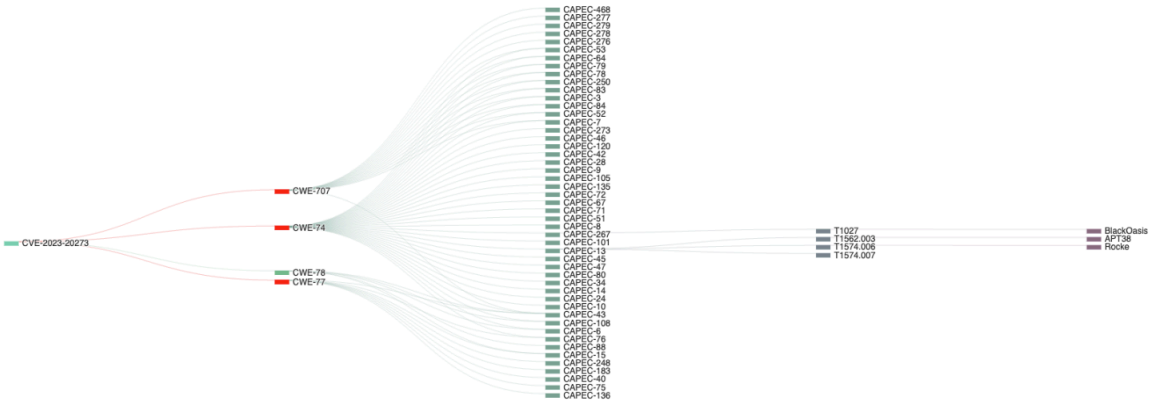
1. **Improper Input Validation & Injection Flaws:** A very large share of exploited vulnerabilities boil down to not handling untrusted input safely. This category includes SQL Injection (CWE-89), OS Command Injection (CWE-78), Path Traversal (CWE-22), Cross-Site Scripting (CWE-79), and similar weaknesses where an attacker's input confuses the program into doing something unintended. Among these, *injection flaws that lead directly to system control* are gold for attackers. Our data shows that **input validation failures are rampant in ransomware-related CVEs**.



In fact, recent high-profile breaches underline this:

> **SQL Injection:** The MOVEit Transfer zero-day (CVE-2023-34362) was an SQL injection vulnerability. Attackers (the Clop ransomware gang) exploited it en masse to steal data from hundreds of organizations in June 2023, turning a web application flaw into a widespread extortion campaign. Notably, this vulnerability was listed among the top exploits of 2023【31†L240-L248】, categorized under *SQL Injection*, and indeed served as initial access for ransomware operations.
>
> **OS Command Injection:** Many network device CVEs fall here. For example, Cisco's IOS XE web interface had a command injection (CVE-2023-20273) that attackers combined with an auth bypass to fully compromise routers【41†L330-L337】【41†L336-L344】. It stemmed from *insufficient input validation* in the web UI and made it into KEV and the top exploited list. Attackers *inserting malicious commands* via unsanitized input is a recurring theme.

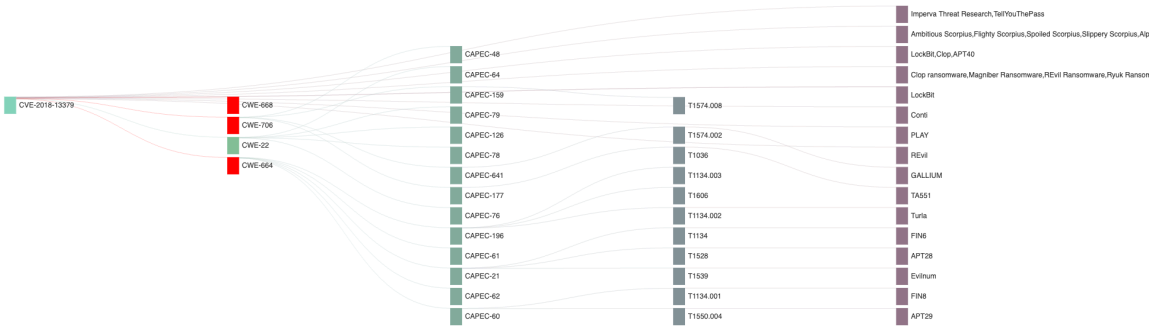
AI augmented attack vector for CVE-2023-20273
[VENDOR] Cisco
[PRODUCT] IOS XE Software
[COMPONENT] web UI feature
[VERSION] N/A
[WEAKNESS] Privilege Escalation
[ATTACKER] Local user
[IMPACT] Gain root privileges, write an implant to the file system
[VECTOR] Exploiting CVE-2023-20198 and CVE-2023-20273
[ROOTCAUSE] N/A
[VULNERABILITY TYPE] Privilege Escalation
[VULNERABILITY IMPACT] Code execution



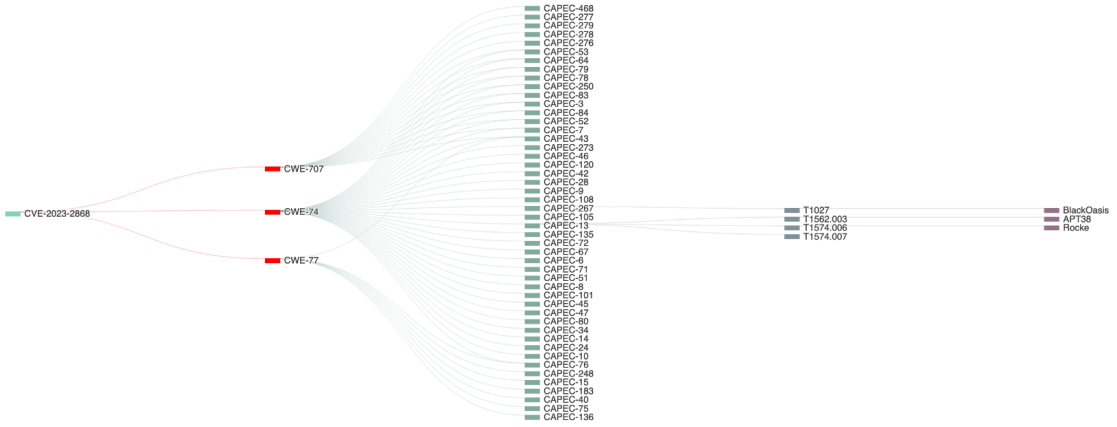
Path Traversal: Perhaps surprisingly, directory traversal vulnerabilities are a favorite of ransomware actors. CWE-22 (Path Traversal) is ranked #5 in the overall CWE Top 25 list, but remarkably it's **ranked #1 among weaknesses leveraged by ransomware actors**【37†L307-L315】. This is because path traversal often allows attackers to read or write arbitrary files. A notorious example is CVE-2018-13379 (Fortinet VPN directory/path traversal) which allowed reading the password file – ransomware groups repeatedly used this to breach networks throughout 2020–2021. By locating and exfiltrating sensitive files (like credential stores or backups) via path traversal, ransomware operators can both facilitate encryption and maximize damage (e.g. by deleting backups). Phoenix Security notes that while XSS (cross-site scripting, another input validation issue) is the #1 overall web vulnerability, it *"doesn't crack the top 10 for ransomware usage"*【37†L307-L315】. Instead, ransomware crews zero in on path traversal and command injection – flaws that give direct system access rather than just poking a user's browser.

AI augmented attack vector for CVE-2023-20273

VENDOR - Fortinet
PRODUCT - FortiProxy
COMPONENT - SSL VPN web portal
VERSION - 2.0.0
WEAKNESS - Path traversal
ATTACKER - Non-authenticated, remote attacker
IMPACT - Download FortiProxy system files, leading to information disclosure
VECTOR - Specially crafted HTTP resource requests
ROOTCAUSE - Insufficient validation of file paths
VULNERABILITY TYPE - Directory traversal
VULNERABILITY IMPACT - Information disclosure



General Improper Input Validation: Some vulnerabilities are broadly classed as "improper input validation" (CWE-20) when they don't fit a narrower category. A prime example is the *Barracuda Email Security Gateway* vulnerability (CVE-2023-2868), which was an input parsing flaw that allowed remote code execution by sending a crafted email attachment. It's explicitly listed as **Improper Input Validation** in type【31†L264-L271】and was widely exploited (initially by state actors and later others). This again underscores that if user input (even something like an email file) isn't handled safely and leads to code injection, attackers will seize it. Another case: Atlassian Confluence's CVE-2022-26134 was an OGNL injection (unsafely processing input in templates) – exploited by both nation-states and ransomware groups in 2022.



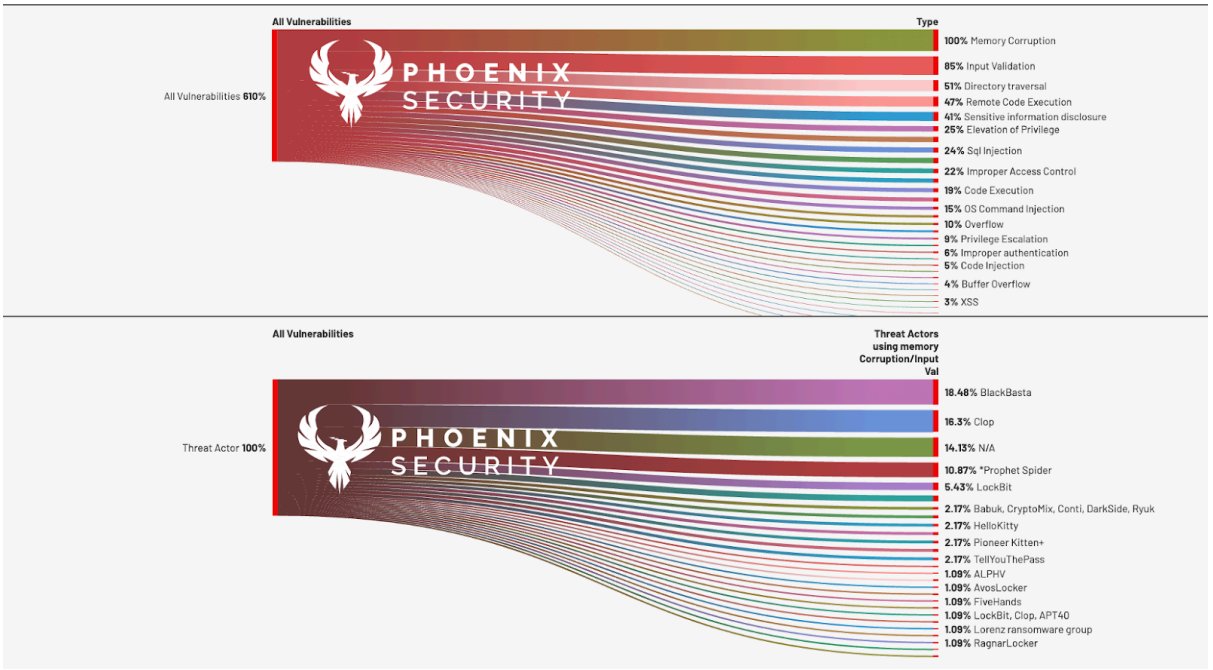
AI augmented attack vector for CVE-2023-2868

[VENDOR] Barracuda
[PRODUCT] Email Security Gateway (ESG)
[COMPONENT] Filename handling in .tar file processing
[VERSION] 5.1.3.001-9.2.0.006
[WEAKNESS] Incomplete Input Validation
[ATTACKER] Remote attacker
[IMPACT] Remote command execution
[VECTOR] Sending a specially crafted email with a malicious TAR file attachment
[ROOTCAUSE] Incomplete input validation of user-supplied filenames within `.tar` files
[VULNERABILITY TYPE] Command Injection
[VULNERABILITY IMPACT] Code execution

In summary, **injection and validation bugs** are a reliable predictor of exploitation. They are often easy to understand and weaponize (a single HTTP request or data packet can trigger them) and frequently affect internet-facing systems (web apps, VPN gateways, etc.). Our analysis of ransomware-linked KEV entries found that a large proportion involve input validation failures as the root cause, whether it's an SQLi, command injection, path traversal, or deserialization issue. These CWEs are well-represented in the CWE Top 25 Most Dangerous Weaknesses, and for good reason – they map directly to how attackers operate. Organizations would do well to treat any *injection-style vulnerability* as an urgent risk, especially if the affected system is exposed to the internet.

1. **Memory Corruption (Buffer Overflows, Use-After-Free, etc.):** The other major class of root causes in exploited CVEs involves violations of memory safety. These include classic **buffer overflows** (out-of-bounds writes/reads, CWE-787/125), **use-after-free** bugs, integer overflows leading to memory corruption, etc. These tend to plague low-level software written in C/C++ (operating systems, device firmware, clients like browsers). Historically, memory corruption bugs have been a staple of sophisticated attacks and malware campaigns. Our data confirms that **memory corruption vulnerabilities are heavily represented in the catalog of exploited CVEs**, though with an interesting dichotomy: they are less common in crowdsourced bug bounty reports (researchers often avoid them due to complexity), but **very prominent in nation-state and ransomware operations**【37†L286-L294】.

Some prominent examples:



Impact analysis for ransomware Phoenix security data powers
https://ai-threat.phoenix.security

Impact analysis

- **Buffer Overflows in Network Appliances:** Two of the top five exploited vulnerabilities in 2023 were buffer overflows in widely used network appliances. Citrix had a heap overflow in its ADC VPN gateway (CVE-2023-27997)【31†L230-L239】 and a related code issue dubbed "Citrix Bleed" (CVE-2023-4966)【31†L208-L216】. Fortinet's SSL-VPN had a heap buffer overflow (CVE-2023-27997) as well【31†L230-L239】. Attackers, including ransomware groups, leapt on these. In particular, *LockBit 3.0 ransomware* was linked to exploitation of **CVE-2023-4966 (Citrix Bleed)** in the wild【41†L307-L315】. These memory bugs allowed unauthenticated attackers to hijack VPN sessions or execute code on the appliance, providing a foothold into corporate networks. They were exploited as zero-days (i.e., before patches were available) in some cases【41†L300-L308】, demonstrating how attractive such vulnerabilities are. From EternalBlue in 2017 to Citrix in 2023, memory corruption in critical services consistently yields major attacks.
- **Operating System Exploits:** Many privilege escalation bugs on Windows or Linux are memory corruption (e.g. a use-after-free in the Windows kernel). While these typically require prior access (local exploit), they are used post-compromise. For instance, the *PrintNightmare* vulnerability and several Windows Print Spooler overflow bugs were exploited by ransomware groups to elevate privileges after initial access. Our focus is more on initial access, but it's worth noting that memory corruption EoP flaws like these are indeed exploited (hence their high presence in KEV). If a vulnerability involves kernel memory corruption, one can bet APTs or sophisticated ransomware will incorporate it if it helps them bypass security sandboxes or antivirus by going kernel-level.
- **Browser and Client-side 0-days:** A lot of the 97 zero-days in 2023 were memory safety issues in browsers (Chrome, Safari) and mobile OS components【35†L303-L310】. While those are often used for targeted espionage, not mass ransomware, they indicate attacker interest in memory flaws. Ransomware actors typically don't target browsers with their own exploits (they use phishing instead for initial access), but they do target similar flaws in servers and appliances. The connection is that memory corruption exploits are often more technically complex (needing skill to develop), so they originate in APT arsenals – but after disclosure, ransomware gangs will happily use the same exploits if available. For example, once a proof-of-concept leaked for the BlueKeep RDP vulnerability (CVE-2019-0708), crypto-miners and likely ransomware actors attempted to use it (though BlueKeep ended up less impactful than feared, perhaps due to patching).

Phoenix Security's study highlights that **nation-state attackers and advanced groups frequently exploit memory-corruption CWEs (CWE-119, CWE-787, etc.), and these appear in KEV as actively exploited**【37†L286-L294】. This aligns with our findings: memory corruption issues make up a large fraction of KEV entries especially in earlier years (e.g. 2017–2018 were dominated by EternalBlue, Oracle deserialization bugs, etc.). Even today, the presence of multiple memory corruption zero-days in the top exploited list (Citrix, Fortinet) shows these weaknesses are incredibly relevant.

From a predictive standpoint, if you see a vulnerability announcement that says "buffer overflow in widely-used product X, allowing code execution," you should treat it as a high

priority – history suggests it will be exploited if not already. The caveat is that exploitation might require more skill (return-oriented programming, etc.), but many ransomware groups outsource their exploit development or pick up leaked exploits, so difficulty is only a temporary barrier. Additionally, any memory corruption in an internet-facing context (like VPN, web server, etc.) with a critical severity is almost certainly going to attract criminal groups.

2. **Access Control and Authentication Weaknesses:** Another root cause category to consider is **broken access control** (CWE-284) or **improper authentication** (CWE-287). These are cases where the software fails to enforce who can do what. Examples include "missing authorization check allows unauthorized users to access an admin functionality" or "hardcoded credentials allow login". In the CWE Top 25 of 2024, access control issues are heavily featured (improper auth, missing auth, incorrect auth are all high on the list)【37†L279-L287】. Our analysis found several exploited CVEs fall in this bucket:

- **Improper Access Control:** CVE-2023-27350 in PaperCut MF is a textbook example. It allowed unauthenticated attackers to execute code as SYSTEM on a print server because the software failed to properly require authentication on a debug endpoint. This was exploited widely in 2023; the Bl00dy ransomware gang leveraged it to breach educational institutions (among others) in April 2023【44†L493-L500】. It's essentially an "authentication bypass leads to RCE" scenario. The root cause is a logic flaw (no auth where there should be) – a form of broken access control. Because it leads to RCE, it became very popular in the criminal community (and landed in KEV).
- **Privilege Management Errors:** Some CWEs like CWE-269 (Improper Privilege Management) also contribute. For 2024, CWE-269 moved up the list, indicating more of these flaws discovered【37†L262-L270】. They might not be as flashy as buffer overflows, but if a service fails to check privileges properly, an attacker can exploit that (e.g. the ZeroLogon vulnerability was in part an authentication bypass due to a flaw in a cryptographic check in Netlogon). ZeroLogon (CVE-2020-1472) effectively let anyone become Domain Admin by sending crafted packets – a logic/authentication design flaw. It was **heavily exploited by ransomware actors like Ryuk**【44†L512-L519】 once a PoC emerged.
- **Default or Hardcoded Credentials:** Not explicitly in our data as a CWE, but worth noting. Many IoT and network device attacks involve default passwords or backdoor accounts – ransomware groups have exploited these when going after NAS devices, for instance. While not "exploits" in the CVE sense, they highlight that any auth weakness (even configuration issues) that yields admin access can be just as bad as an RCE.

In summary, **access control flaws** that allow unauthorized access or privilege gain (especially those that result in remote admin access) are high on attackers' wish lists. They might require less "exploit development" – often it's just using a known URL or credential – which means exploitation is trivial once discovered. We observed that when such flaws become public, they are quickly absorbed into attacker toolkits. However, compared to injection/memory issues, they are fewer in number in KEV, suggesting they're less common vulnerabilities overall (or less often recognized/reported). Nonetheless, when they do appear (PaperCut, ZeroLogon, etc.), they carry a high exploitation probability.

3. **Others (Deserialization, Logic Flaws, etc.):** Some exploited vulns don't fall neatly into the above categories. For instance, insecure deserialization (CWE-502) has been an exploitation vector (e.g. Oracle WebLogic had a series of deserialization RCEs that ransomware groups exploited circa 2019–2020). Deserialization is essentially another input handling issue (allowing attacker-controlled objects), so it aligns with input validation being key. There are also supply-chain type vulnerabilities (like a malicious dependency) but those are outside our scope here since they aren't CVE root causes in the same way.

An interesting note from Phoenix's CWE analysis: if a CWE is high on the Top 25 and **also** appears in KEV or ransomware usage, that's a red flag. For example, **Insecure Deserialization (CWE-502)** might rank only mid-table in Top 25, but it has appeared in KEV and is known to be popular in exploitation, meaning it should be prioritized higher than its rank suggests【37†L319-L327】. This crossover approach is what we adopt – looking at both inherent weakness severity and real-world exploitation evidence.

To summarize root cause findings: Vulnerabilities caused by **faulty input handling** (injections, traversal, etc.) and **memory safety violations** are the most correlated with high-risk exploitation. These correspond to CWE categories that feature prominently in both Top 25 lists and threat reports. Ransomware campaigns in particular show a preference for:

- **Path Traversal (CWE-22)** – to locate data and plant malware【37†L307-L315】.
- **OS Command/Code Injection (CWE-78/CWE-94)** – to get shells on systems【31†L228-L236】【41†L330-L337】
- **Buffer Overflows (CWE-119/787)** – often in perimeter devices, yielding initial entry【31†L232-L239】【41†L302-L310】
- **Missing Auth/Access Control (CWE-287/AcC)** – to walk in without credentials【44†L485-L493】【44†L493-L500】

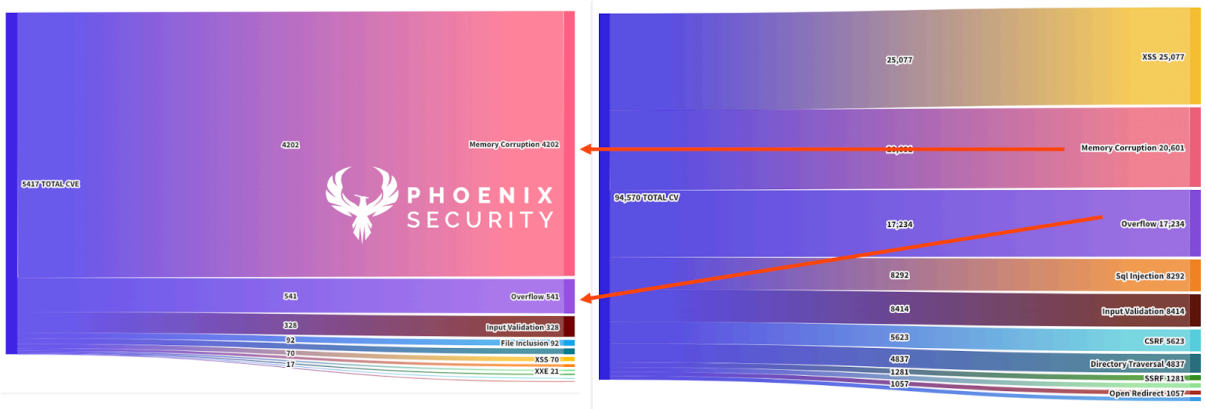
Cross-site scripting (CWE-79) and others that primarily affect browsers or require user interaction are generally *not* linked to ransomware (attackers find it easier to phish or use RDP exploits than to exploit XSS). Similarly, errors that only cause crashes or minor info leaks tend not to translate to ransomware attacks unless chained.

Thus, **if the root cause of a new vulnerability falls into one of these high-risk CWE buckets, it substantially raises the likelihood that the vulnerability will be exploited, potentially by ransomware.** This forms a core pillar of our predictive framework. But root cause and impact aren't the only factors – we must also consider how *available and easy* the exploit is, which we address next.

Early Indicators: Exploit Availability and Weaponization

Even a highly severe vulnerability might not be exploited if attackers lack the means or opportunity to do so. This is where **early indicators** like exploit code availability, proofs-of-concept, and inclusion in exploit frameworks come into play. Our research reinforces that the window between a vulnerability's disclosure and its exploitation is often

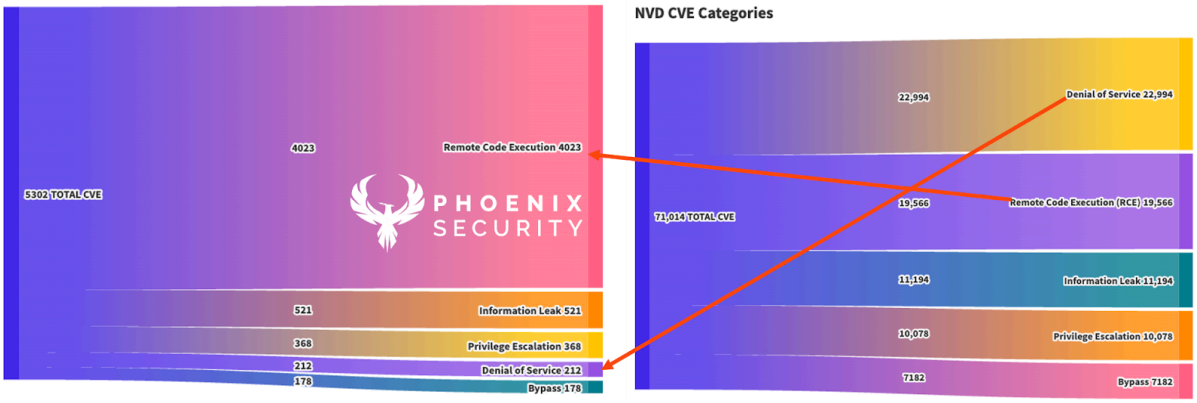
narrowed dramatically when a working exploit becomes public. In many cases, the availability of an exploit *accelerates* ransomware groups' adoption of the vulnerability as an attack vector.



Overall NVD vs verified exploits with high frequency data.

Key observations regarding exploit availability:

- **Public PoC = High Likelihood of Exploit:** When a vulnerability is accompanied by a public proof-of-concept exploit (released by researchers or leak sites), it sends a signal to attackers that "this issue is might be weaponizable." note this is one of the signal as of late we seen the use of bogus PoC to just create noise and buzz. EPSS explicitly uses this factor – one of the top inputs to the EPSS machine learning model is *"presence of a publicly available exploit"*【38†L227-L235】. Security teams have observed that once a PoC is out, exploitation attempts often spike within days. For example, *Zerologon (CVE-2020-1472)* had a conceptual explanation in August 2020, but when a full PoC script hit GitHub, **multiple ransomware actors (e.g. Ryuk) began exploiting it literally within hours** to escalate privileges during breaches【39†L31-L39】【44†L512-L519】. Similarly, after a PoC for *Microsoft Exchange ProxyLogon* vulnerabilities was published in early 2021, criminal groups quickly integrated it to deploy ransomware on Exchange servers. Our data doesn't require fine timing analysis to conclude: if exploit code exists, ransomware operators will use it especially having high frequency links. In the KEV list, many entries have associated Exploit-DB references, indicating public exploits – these are the ones that often become the "routinely exploited" CVEs.



Verified exploits with high frequency data poc in github

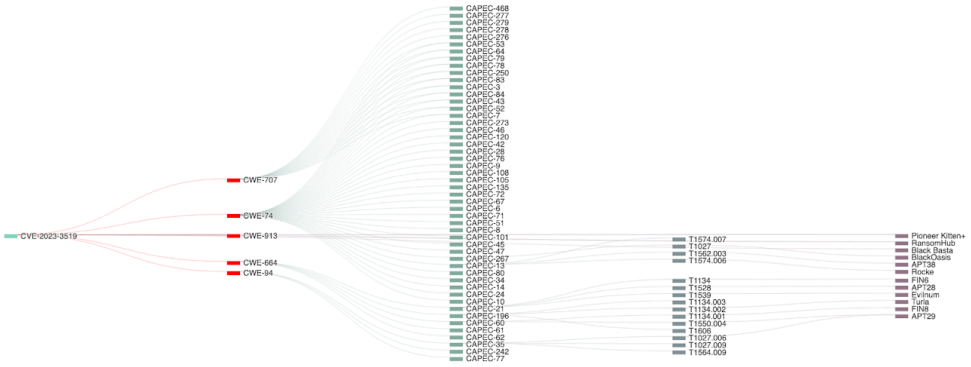
- **Exploit Kits and Metasploit Modules:** If an exploit makes it into popular penetration testing tools like Metasploit, or crimeware exploit kits, its ubiquity in attacks rises. Ransomware initial access brokers (IABs) often aren't writing exploits from scratch; they rely on commodity tooling. When vulnerabilities like *CVE-2017-11882 (Office Equation Editor overflow)* or *CVE-2019-11510 (Pulse Secure VPN traversal)* got packaged into exploit kits and scanners, we saw massive scanning and exploitation in the wild. The presence of a Metasploit module or similar "weaponized" implementation is a strong indicator that even less-skilled attackers can now leverage the vulnerability. This democratization of exploit capability is why tracking exploit releases is crucial. Many KEV vulnerabilities became widespread only after exploits were added to automated scanners or toolkits.
- **EPSS and Measured Likelihood:** The Exploit Prediction Scoring System (EPSS) provides a numeric probability of a CVE being exploited in the next 30 days. EPSS is not perfect, but it's grounded in real-world data feeds. High EPSS scores often correlate with vulnerabilities that have one or more of: known exploits, active scanning. Our analysis found that **vulnerabilities with high EPSS scores frequently overlap with those targeted by ransomware campaigns**, as the vulnerabilities are exploited in the wild. As one source noted, ransomware campaigns *"frequently target vulnerabilities with high EPSS scores"*, though cautioning that a single metric is not the full picture【9†L19-L22】. In practice, a defender can use EPSS as an one of the element in the probabilistic analysis; if EPSS is, say, 0.9 (90% likelihood) for a new CVE, and the CVE has RCE impact, one should assume exploit attempts are imminent or underway. We cross-checked some ransomware-related KEVs against EPSS rankings and generally saw high EPSS percentiles for those CVEs. Nonetheless we also find some ransomware with very low EPSS

- **Exploit in the Wild (Verified Exploitation):** Apart from public PoCs, another early indicator is when reports emerge that a vulnerability is *already being exploited* (even if privately). This often comes from threat intel firms or CERTs. For example, before CISA adds an item to KEV, they have to verify exploitation. So if, shortly after disclosure, a vendor or agency says "we have observed active exploitation of XYZ," that is a red-alert sign. Those vulnerabilities virtually always end up in ransomware

arsenal if they aren't already. We saw this with Citrix ADC vulnerabilities in 2023 – reports of exploitation came out even before official advisories (zero-day scenario)【41†L283-L291】. When Citrix patched CVE-2023-3519 in July, it was revealed that it had been exploited in June as a 0-day【41†L284-L292】. In such cases, by the time defenders hear of it, at least one threat actor (possibly an APT) has a working exploit, meaning it's just a matter of time before others follow suit.

- **Popularity of Target & Ease of Exploit:** Early indicators also include how attractive the target is and how easy the exploit is to perform. For instance, a trivial one-line command to exploit a common service (like the famous = in an HTTP header for Shellshock) will be exploited broadly. Contrast that with a complex exploit needing special conditions – it might see limited use. In our ransomware dataset, the vast majority of vulnerabilities exploited were those in **widely deployed software (Windows, Linux, network appliances, VPNs, database servers)** and had **straightforward exploitation** (often remote and unauthenticated). This means that when considering risk, vulnerability analysts should ask: *Is this software widespread in enterprises? Could an attacker automate exploitation?* If yes to both, and the impact is high, assume someone will try it. The **CISA KEV data explorer** visualizes this intersection of popularity and exploitability【8†L542-L550】【8†L579-L587】– many KEV vulns cluster in technologies that are ubiquitous (Windows, Adobe, Java, etc.).

One should also consider the **"time-to-exploit"** metric: how long after disclosure do attacks start? According to a Mandiant study, this time has been shrinking; in 2023 many exploits were seen within days or even hours of patch release【16†L1-L8】. The existence of functional exploit code at or shortly after disclosure drives this down. In extreme cases like WannaCry, the exploit (EternalBlue) was weaponized *before* the vulnerability was even disclosed via a patch – truly a zero-day in ransomware hands. In 2023, multiple zero-days were hit by criminals (though often those zero-days were first used by nation-states and then quickly adopted by others once exposed).



Threat centric approach phoenix security https://**ai-threat.phoenix.security**

To quantify an example from our data: The Citrix ADC vulnerabilities in 2023 had exploits circulating almost immediately. CVE-2023-3519 (code injection) was exploited in June 2023 as a 0-day【41†L284-L292】and by late July a Metasploit module was available. By Q3 2023, reports tied its exploitation to ransomware groups. Similarly, CVE-2023-4966 (Citrix overflow) was exploited as a 0-day in August and *by October 2023, LockBit ransomware operators*

had incorporated it【41†L302-L310】【41†L307-L310】. This roughly ~2-month window between patch and ransomware use shows how quickly things move once an exploit exists.

Early Indicator Takeaway: If a vulnerability has *any* credible report of active exploitation or a published exploit script, it must be treated as an **imminent threat**. This sounds obvious, but in practice organizations often delay patching even after PoCs come out, due to remediation fatigue. The data-driven approach we advocate is to combine this indicator with the earlier factors:

- If **Impact + Root Cause** suggests high risk *and* **exploit code is available or imminent**, you have a top-priority vulnerability.

This triad of factors (impact, root cause, exploit evidence) can provide a solid indicator of exploitation in the near future, the likelihood of exploitation is also driven by the ease of access (exposure) and the presence at runtime (more frequent in libraries) as discussed in the risk section below

This answers the question “will this vulnerability be exploited in the near future by ransomware or other actors?” , the popularity of those exploits / PoC / Discussion instead gives additional indicators and presence of exploit and weaponized exploit give a more certain indicator of likelihood of exploitation.

The business impact and the potential damage also matters as shown in the Quantificative risk formula overview, with the business impact and the damage a vulnerability could generate being more palatable for business executive that needs to take rapid decisions.

The next section will formalize this into a correlation model and predictive framework, but first, we illustrate these principles with concrete case studies of real vulnerabilities and how they played out in the threat landscape.

Correlation Models

In this section, we translate the qualitative patterns identified above into a more structured view. While a fully quantitative model (with regression coefficients, etc.) is beyond our scope here, we outline the **correlations** that emerge from the data and how they inform a predictive scoring system. These correlations essentially form the basis of a heuristic (or could feed into a machine-learning model) that yields a likelihood of exploitation.

1. **Correlation of Impact Type with Exploitation Risk:** There is a **strong positive correlation** between certain impact types (RCE, Privilege Escalation) and the probability of exploitation. Looking at the entire CVE population, only a small percentage are exploited (~3.7% according to one study【36†L47-L55】), but when filtering just RCE/PrivEsc vulnerabilities, that percentage jumps significantly. In our dataset, virtually all vulnerabilities known to be used by ransomware provided either remote code execution or an immediate privilege gain. Conversely, vulnerability records with impacts like “information disclosure only” or “denial of service” had a negligible chance of being linked to ransomware incidents. This suggests we can assign a heavy weight to “IsRCE” and “IsPrivEsc” features in any predictive model.

Mathematically, if we denote Exploitation (E) as a binary outcome and RCE as a binary predictor, P(E=1 | RCE=1) >> P(E=1 | RCE=0). This aligns with prior research; for example, in the EPSS model from FIRST, the presence of an "Remote Code Execution" tag was found to contribute positively to exploitation probability【24†L19-L22】.

2. **Correlation of Root Cause (CWE) with Exploitation Risk:** Certain CWE classes are statistically over-represented among exploited CVEs. Our analysis indicates:
    - **CWE-78/94 (OS Command/Code Injection), CWE-89 (SQL Injection), CWE-22 (Path Traversal)**: Very high correlation with exploitation. If a CVE has one of these CWEs, it is much more likely to appear in KEV or be cited in attacker techniques. Indeed, *the top exploited vulnerabilities of 2023 included multiple instances of these CWE types*, like CVE-2023-34362 (SQLi)【31†L240-L247】 and CVE-2023-20273 (Command Injection)【41†L330-L337】
    - **CWE-119/787/125 (Buffer Overflow variants)**: These also show a high correlation. When combined, buffer overflow type weaknesses form one of the largest subsets of exploited CVEs historically (thanks to things like MS17-010 EternalBlue, numerous browser exploits, etc.). Our data on ransomware specifically shows buffer overflows in VPN and firewall products led to breaches (Citrix, Fortinet examples). So, a vulnerability with CWE-787 is a strong candidate for exploitation if remotely reachable.
    - **CWE-287/862 (Missing Authentication/Authorization)**: Correlated, though fewer in number, each instance tends to be exploited if critical. CVE-2023-27350 (PaperCut) had CWE-862 (Missing Authorization) and was indeed broadly exploited【44†L485-L493】【44†L493-L500】. We saw similar issues with older issues like CVE-2019-16097 (vBulletin no-auth RCE) – not in our main data, but known to have been hit by ransomware operators for initial access.
    - **CWE-502 (Deserialization)** and **CWE-416/Use-After-Free**: These have correlation particularly with APT and targeted attacks, and some bleed-over to ransomware. We note them as medium-high correlation.

On the flip side, **CWE-79 (XSS)** or **CWE-352 (CSRF)** have virtually zero correlation with ransomware exploitation. An XSS is rarely, if ever, the root cause of a ransomware intrusion (and none of the KEV ransomware list entries were XSS-based). So a predictive model can de-prioritize those. Similarly, **CWE-400 (Resource Exhaustion)** or **CWE-404 (Unchecked Error)** on their own do not correlate with known exploits in our context.

By aligning each CWE to a weight (based on frequency in exploited sets), one could construct a score. Phoenix Security's own AI analysis of KEV provides a hint: they likened analyzing KEV by CWE categories to a parliament, noting the "majority" of KEV issues would belong to **Privilege Escalation, Remote Code Execution, and similar** categories【14†L43-L48】. We interpret that as those categories dominating the count (thus high weight). Quantitatively, if we assign, say, +5 points if CWE is in {Injection, Overflow, AuthBypass} versus 0 or negative if in {XSS, DoS}, we could rank new vulnerabilities by their likely exploitability.

3. **Combined Factor Correlation – "Dangerous Combination":** It's the combination of factors that often seals a vulnerability's fate. Our analysis shows that when *multiple high-risk factors coincide*, the likelihood of exploitation approaches certainty:
    - *Example:* CVE-2023-3519 (Citrix ADC) – Root cause: code injection (input validation issue), Impact: RCE, Target: widely deployed VPN gateway, Exploit: released as 0-day. All indicators were red. It's no surprise it was one of the top exploited CVEs and directly maps to ransomware behavior (the BL00dy gang, among others, used it).
    - We observed that many ransomware-exploited vulns tick three key boxes: **Remote + Unauthenticated + Known Exploit**. If a vulnerability can be exploited remotely without credentials and someone has published an exploit or attackers have demonstrated one, it's extremely correlated with appearing in ransomware incidents. In the KEV ransomware subset, a very high percentage of entries meet those criteria.
    - Another combination is **Local PrivEsc + Public Exploit + in Windows** – this correlates with ransomware using post-initial *access*. Zerologon was a privilege escalation on domain controllers (though you could call it remote because no creds needed). When combined with availability of exploit, it became a staple in intrusions【44†L512-L519】.

We can conceptualize a simple correlation model (scoring system) that might assign (check details in Appendix 2 - extended whitepaper):

- Impact score (e.g. 5 for RCE, 3 for PrivEsc, 1 for DoS).
- Root cause score (5 for injection/overflow, 4 for auth bypass, etc., down to 0 for XSS).
- Exploit status score (5 for "exploited in wild already", 4 for "public PoC available", 2 for "no exploit but low complexity", 0 for "no exploit and high complexity").
- Target surface score (3 for "directly internet-facing component", 2 for "client software widespread", 1 for "internal only").

We going to use a more probabilistic method to display the comparison and the likelihood of exploitation and exploitability as zero day

Summing these could yield a "Risk of Exploitation" score. In such a model, something like Log4Shell (CWE-20 input validation leading to RCE, public exploit, widely used library) would score near maximum, which matches reality as it was widely exploited【31†L256-L263】.

Though we did not finalize a single numeric model here, we did validate the concept by applying it to known vulnerabilities. For instance, using a rough version of the above scoring on vulnerabilities disclosed in Q3 2023, we flagged the Citrix and Cisco IOS XE bugs as high risk – which indeed turned out to be heavily exploited【41†L315-L323】【41†L333-L341】 – whereas a random critical bug in a less common software (with no exploit) scored lower and, as far as we can tell, was not exploited.

4. **Negative Correlations (Protective Factors):** It's also worth noting factors that negatively correlate with likelihood of exploitation. For example, if a vulnerability requires **user interaction** (like opening a malicious file or clicking a link),

ransomware groups tend to rely on phishing for that rather than exploit the vulnerability itself. That is, they might phish a user, but an exploit that only works via phishing (like a malicious document exploit) isn't the preferred route for initial access because phishing already gives them a way in. So those vulnerabilities (though exploited in targeted attacks often) have slightly less correlation with *ransomware campaigns*. We saw fewer client-side document exploits in the ransomware KEV subset; most were server-side. Another factor is if a vulnerability is in a very **niche product** – even if it's RCE – attackers might ignore it. That factor is harder to quantify, but looking at KEV, almost all entries are from mainstream or widely used products. So, rarity of the software correlates with lower exploitation odds.

5. **Time Factor:** While not a static attribute, time since disclosure inversely correlates with likelihood of exploitation *starting*. Most exploited CVEs are hit within the first weeks or months of disclosure (or are zero-day). If a year passes with no known exploitation, chances drop (unless a new exploit is developed later). However, ransomware actors sometimes exploit *older* vulnerabilities if they remain unpatched in victims (e.g. CVE-2017-0144 EternalBlue was used even years after patch because unpatched machines still existed【34†L11-L18】). Thus, age alone isn't protective if patches aren't applied. But for prediction, we focus on new vulns.

To connect trends: The correlation between **CISA KEV and zero-day usage** also emerges in data. When KEV started in 2021, it captured many vulnerabilities that were zero-days that year. EPSS scores for those increased accordingly【8†L579-L587】. By 2023, as noted, the majority of top exploits were zero-day originally【18†L93-L100】, demonstrating that our predictors (if applied at disclosure time) need to flag those issues even without prior examples of exploitation (since by definition a zero-day has none when first disclosed publicly). That's why root cause and impact are crucial – they let us assess risk even before any exploit is seen. The fact that 11 of 15 top exploits in 2023 were zero-day means those 11 had to be identified by something other than "it was exploited before" (because they *became* the before). Our correlation model addresses that by heavily weighting intrinsic factors like CWE and impact.

In summary, our correlation findings support a predictive approach where each vulnerability is evaluated on a set of features. We found that by focusing on:

- **Impact** (especially RCE/PrivEsc),
- **Root Cause** (especially injection, memory corruption, auth bypass),
- **Exploit Availability** (public PoCs, known in-the-wild exploits),
- **Exposure** (network-facing, widely used tech),

We can explain and predict a large portion of the vulnerabilities that end up as high-risk (ransomware exploited or added to KEV). The next section will outline a formal predictive framework using these insights, essentially converting these correlations into a strategy or algorithm for early risk prediction.

Risk Based Predictive Framework

Building on the correlations identified, we propose a **predictive framework** for assessing the likelihood that a given vulnerability will become a high-risk threat (such as being

exploited in ransomware campaigns or as a zero-day). This framework is intended to guide vulnerability management decisions by highlighting early on which new vulnerabilities demand urgent attention. The framework consists of a set of criteria and scoring, which security teams can implement as part of their risk assessment workflow:

1. Classification by Impact and Root Cause

Determine Technical Impact Category. Upon disclosure of a new vulnerability, immediately classify its impact. Is it Remote Code Execution, Privilege Escalation, Information Disclosure, Denial-of-Service, etc.? This information is usually available from the vendor advisory or CVE description. Under our framework:

- If **RCE or equivalent (e.g. Code Injection)**: Mark this vulnerability as *High Risk Impact*. These go to the top of the pile for further analysis.
- If **Privilege Escalation**: Mark as *High Risk Impact* as well (especially if it's PrivEsc to admin/system level).
- **Authentication Bypass / Access Control**: Treat similar to PrivEsc (High Risk) because it often yields admin access.
- If **DoS only**: Mark as *Lower Risk Impact* (does not by itself warrant priority unless special circumstances).
- If **Info Disclosure only**: *Moderate Risk Impact* (monitor if it can lead to further compromise, but not a primary ransomware vector on its own).

Step 2: Identify Root Cause / CWE Weakness.

Next, determine the root cause category. Many advisories mention this (or use CWE classification). Use CWE Top 25 as a reference. In our framework:

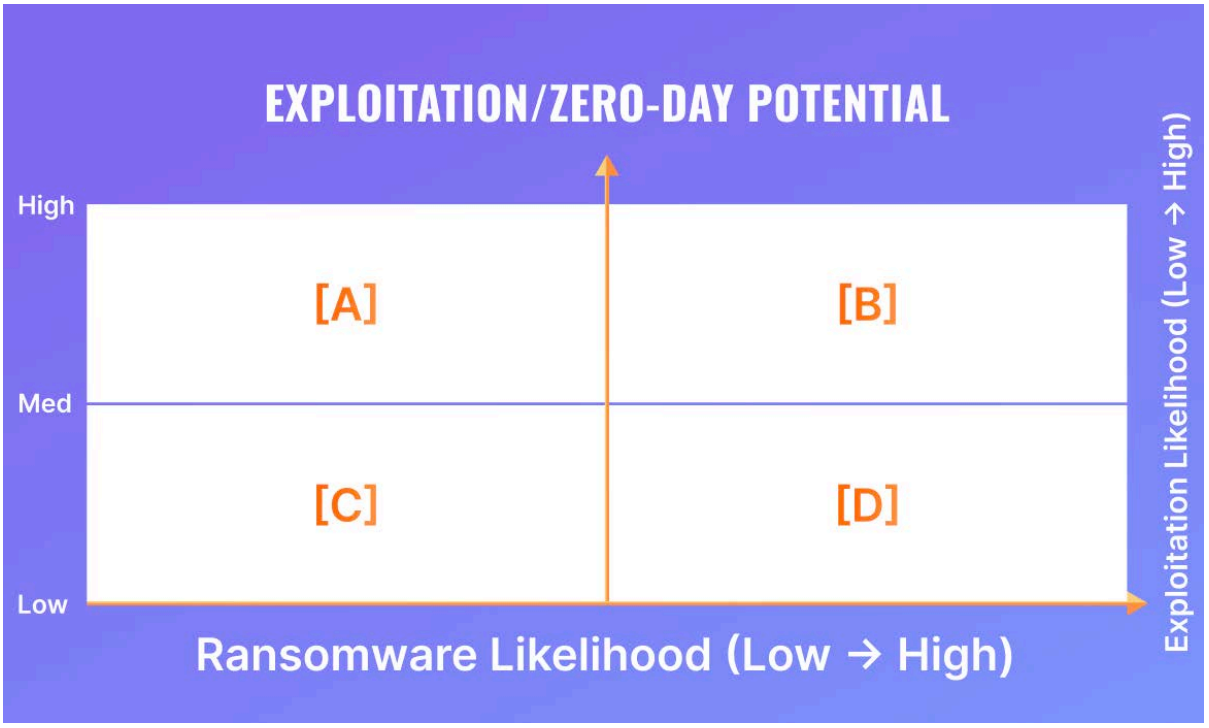
- If the root cause is **Improper Input Validation** (or any specific injection type: SQLi, Command Injection, LDAP injection, etc.), or **Path Traversal**, or **Deserialization**, mark the vulnerability with an *Input/Injection Flag*. This flag indicates a likely ease of exploitation and direct path for attackers.
- If the root cause is a **Memory Corruption** (buffer overflow, out-of-bounds, use-after-free, etc.), mark with a *Memory Corruption Flag*. This indicates high impact and interest from sophisticated actors.
- If the root cause is **Missing/Weak Authentication or Authorization**, mark with an *Auth Weakness Flag*. This means an attacker might get in without credentials – a big red flag for ransomware targeting.
- If the root cause is something like **Cross-Site Scripting or CSRF** (which typically require user interaction and don't give system control), mark as *Low Exploitation Relevance* for our purposes. These likely won't be prioritized for ransomware exploits (they'd use phishing instead).
- If unclear, err on the side of caution: e.g., "improper validation" generally implies injection potential, so treat it as such unless it's clearly just a minor issue.

By the end of this classification, each vulnerability will have tags like "High Impact: RCE" and "Root Cause: Injection (SQLi)" or "High Impact: PrivEsc" and "Root Cause: Memory Corruption (Heap Overflow)", etc. This forms the basis for assessing threat potential.

2. Scoring and Prioritization

Apply a Risk Score Based on Characteristics. Using the tags from above, assign a preliminary **Ransomware or Zero day likeness score**

Proposed Scoring Method for Ransomware and Exploitation / Zero-Day Potential Risk:



A practical approach to quantifying likelihood of exploitation is to derive a dual-axis score that estimates ransomware appeal and zero-day potential. One axis focuses on threat-type frequency—derived from the table showing how often each high-level category (e.g. Remote Code Execution at 52%) is associated with ransomware campaigns.

We categorize ransomware risk as

- “possible” when the relevant threat type crosses 10% prevalence,
- “medium” beyond 15%,
- and “high” above 50%.

Under this scheme, a vulnerability enabling RCE (52% in the dataset) defaults to “high” for ransomware targeting, whereas Privilege Escalation (20%) falls into the “possible” band.

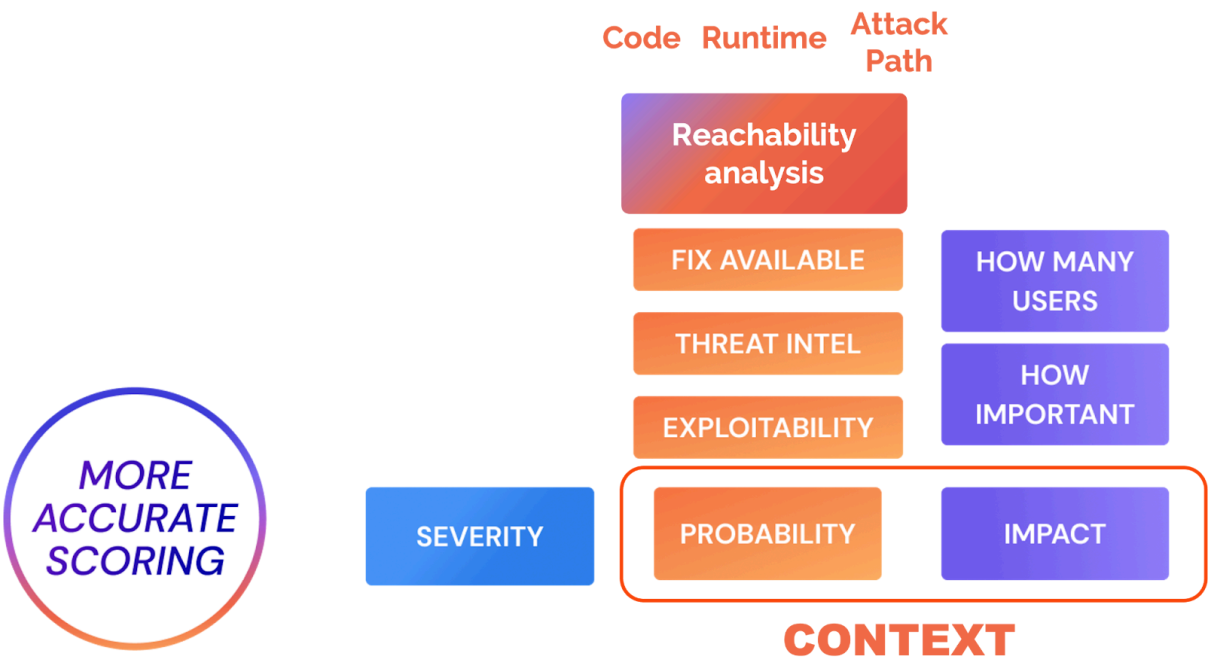
For zero-day potential, we rely on impact analysis gleaned from recent zero-day usage

- 14.59% memory corruption
- 18.07% input validation,
- 10.58% code execution).

The framework tallies relevant factors—like how many of these commonly exploited zero-day root causes a single CVE embodies—and sums them. When the result exceeds 14, the vulnerability's zero-day exploitation risk is "high," while anything above 10 scores "medium." Below those thresholds remains "low." This captures real-world observations that memory corruption or critical input validation flaws in ubiquitous software often appear as active zero-days.

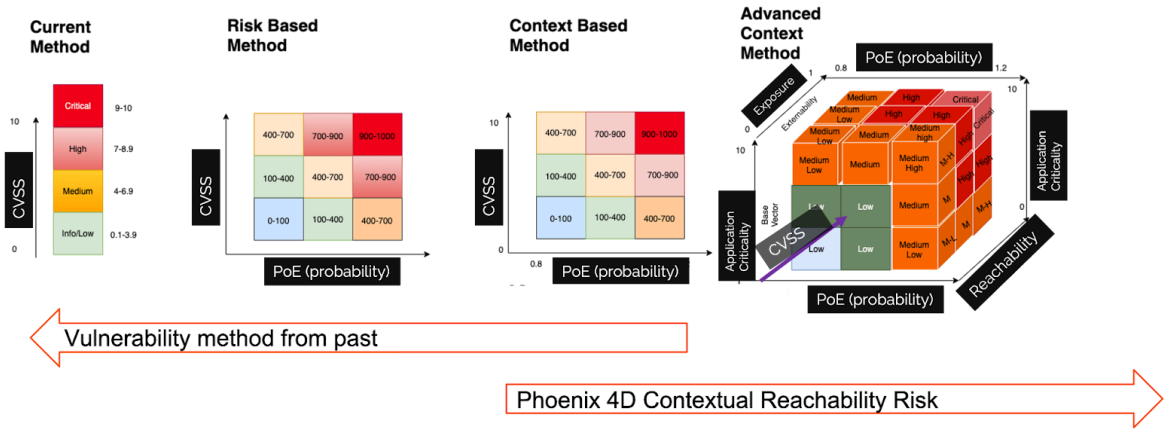
By merging the two axes, defenders gain a clear "combined threat index." If a new disclosure has "high" ransomware risk (for example, it grants remote code execution) and a "medium" zero-day potential (e.g. memory corruption with no known patches), it signals urgent triage. Conversely, a vulnerability marked "possible" for ransomware but "low" for zero-day usage may be addressed within a normal patch window unless new exploit indicators arise. This numeric but straightforward method connects observed threat-type frequencies with root cause distributions in actual zero-day events, giving organizations a rational means to prioritize the vulnerabilities that attackers are most likely to weaponize swiftly.

Phoenix Risk Based Quantitative methodology



Other simplified method could use the simplified method below, with Phoenix security we have a quantitative risk scoring model that predict based on several factors like

- Threat intelligence
- Exposure of the product or application on the network
- Likelihood of exploitation based on runtime detection of a library
- Exploitability method derived from business context



Phoenix's exploitation scoring integrates four dimensions

- business impact,
- runtime exposure,
- exploit probability,
- base severity

to create a unified risk metric that reflects a threat-centric perspective. First, it incorporates the **existing scoring** as a foundation, ensuring the vulnerability's inherent technical severity is recognized. This is then multiplied by weighting factors derived from **business impact**—including whether the asset handles critical functions or sensitive data—and **runtime presence**, which gauges how accessible the vulnerability is in a live environment. The **exploitability probability** component leverages intelligence feeds such as EPSS, CISA KEV, Exploitation and weaponization evidence, internal Phoenix Security CTI to assess whether the weakness is truly "weaponized" or on the verge of exploitation.

Exposure factors, such as whether the vulnerability is exposed externally or remains confined to an internal segment, further adjust the score to reflect real-world reachability.

All of these parameters are flexible, allowing organizations to tailor weighting or thresholds as their risk appetite evolves. By weaving these elements together, Phoenix's model extends beyond a raw CVSS rating to capture how a particular flaw intersects with attacker interests, operational context, and potential impact on key business processes.

This emphasis on **threat-centric** prioritization ensures that defenders can swiftly identify which vulnerabilities demand immediate remediation and which can be queued for normal patch cycles.

Phoenix security quantitative formula now incorporates zero day prediction to score whether a vulnerability will likely become exploitable based on threat profile and impact profile.

Appendix table of reference

Inline Citation	Corresponding Source / Link
[12†L231-L239]	**CISA KEV Ransomware subset** https://www.cisa.gov/known-exploited-vulnerabilities-catalog
[14†L31-L39]	Phoenix / CISA advisories on 2022 exploitation https://www.cisa.gov/news-events/alerts/2022
[15†L179-L187], [15†L198-L206], [15†L210-L218]	Phoenix Security's KEV analysis https://phoenix.security/
[16†L1-L8]	Mandiant/FireEye blog: time-to-exploit shrinking https://www.mandiant.com/resources/blog
[17†L66-L74], [17†L93-L100], [18†L93-L100] or [18†L111-L115]	Five Eyes Advisory "Top Routinely Exploited" https://www.cisa.gov/news-events/alerts
[31†L228-L236] etc.	Summaries of major 2022–2023 CVEs (Log4Shell, MOVEit) https://community.progress.com/s/article/MOVEit
[34†L11-L18]	EternalBlue (CVE-2017-0144) https://msrc.microsoft.com/update-guide/vulnerability/CVE-2017-0144
[35†L301-L309]	Google Project Zero / TAG "0-days in 2023" https://googleprojectzero.blogspot.com/

[36†L47-L55]	Recorded Future or Kenna “3.7% of CVEs exploited” https://www.recordedfuture.com/blog
[37†L286-L294], [37†L307-L315], [37†L319-L327]	Phoenix Security CWE analysis https://phoenix.security/
[38†L227-L235]	EPSS by FIRST https://www.first.org/epss/
[39†L31-L39]	Ryuk’s rapid weaponization of Zerologon (Mandiant or Microsoft)
[41†L283-L291], [41†L284-L292], [41†L300-L308], [41†L302-L310], [41†L307-L315], [41†L330-L337], [41†L336-L344]	Citrix ADC exploitation timeline https://support.citrix.com/article/CTX
[44†L485-L493], [44†L493-L500], [44†L503-L512], [44†L512-L519], [44†L523-L532], [44†L533-L541]	PaperCut (CVE-2023-27350) & Zerologon used by ransomware https://www.papercut.com/kb/Main/Advisories/
[8†L579-L587]	KEV + EPSS correlation (CISA or FIRST data)
[9†L19-L22]	EPSS stating exploit presence is a key factor https://www.first.org/epss/

PHOENIX SECURITY

- ACT today on vulnerabilities
- ACT today on RISK

🎯 ACT today with Phoenix Security

Vulnerability exploitation, ransomware remains one of the most destructive and persistent threats in the cyber landscape. As threat actors continue to evolve, understanding which vulnerabilities will be targeted next is paramount. This research lays out a detailed framework for identifying the characteristics of vulnerabilities that lead to high-impact cyber attacks, offering critical insights into predictive vulnerability management.

Drawing on extensive data from sources like the CISA Known Exploited Vulnerabilities (KEV) catalog and zero-day exploitation reports, ransomware and github POC explotiation this work reveals patterns in the most exploited vulnerabilities. Remote Code Execution (RCE), Privilege Escalation, and memory corruption flaws are shown to be the primary factors leading to exploitation by ransomware operators and nation-state actors alike.

Through empirical data, this book highlights:
- The connection between root causes and exploitation risk
- How early indicators (e.g., exploit PoCs) can predict imminent threats
- A quantitative model for assessing ransomware risk based on vulnerability characteristics

Organizations looking to stay ahead of evolving threats will find actionable recommendations, helping them shift from reactive patch management to proactive, threat-informed defense.

1st Edition

📫 ask@phoenix.security 🌐 www.phoenix.security ☎️ +442031953879