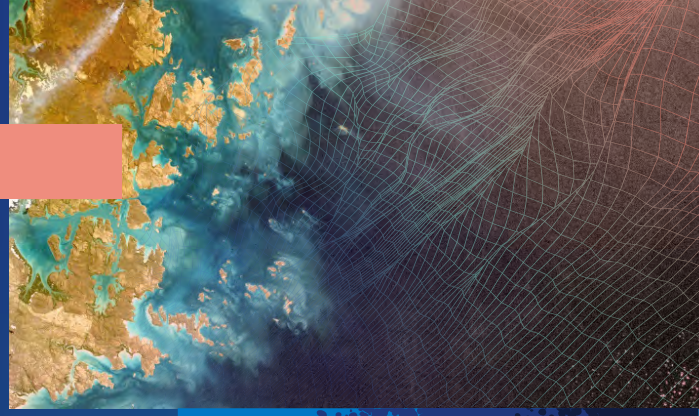
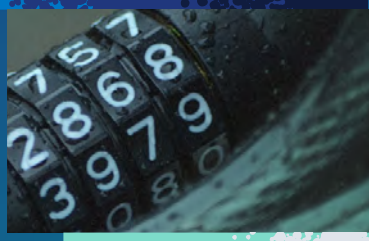
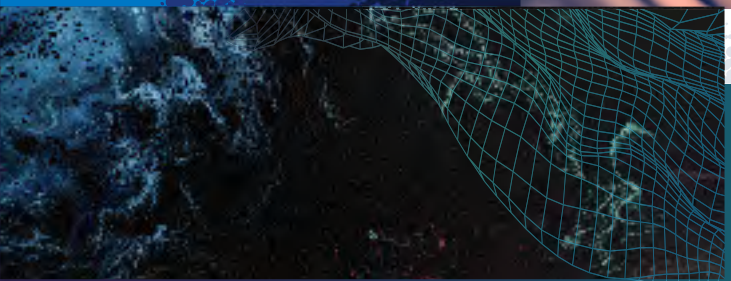




elastic security labs

Global Threat Report

October 2023



Contents

Introduction

The best way to protect the world's data is by weaponizing defensive technologies.

This philosophy has guided Elastic Security Labs since its formation in 2021, and we feel just as strongly about it two years later. Many of our peers in the industry seem content to observe and report on threats, but this permissive posture does little to actually stop cybercriminals. Elastic Security Labs was founded by a group of passionate security leaders who are approaching the problem differently — directly from the front lines.

One step in protecting the world's data involves directly challenging the threat landscape. We do that by democratizing access to knowledge, providing unique insights into threat techniques and tactics, developing and releasing no-cost resources, and leveraging our powerful global instrumentation to return fire. Our efforts make environments hostile to threats and empower organizations to reclaim their environment with the community focused tools we develop.

The last year has seen a massive range of threats from both new and established criminals — delivering an onslaught of attacks designed to leave security teams unaware and overwhelmed. Strategizing for an ever-changing enemy requires an immense amount of planning, and in order to prepare both Elastic and the security intelligence community at large we have constructed the 2023 Elastic Global Threat Report. This report describes threat phenomena, trends, and recommendations we believe will help organizations prepare for the future.

The observations in this report are based on Elastic telemetry, public, and third-party data that has been voluntarily submitted. We welcome the opportunity to partner with our customers in this way to analyze their data, anonymously sharing what we learn with the larger security industry. All information has been enriched with cutting-edge innovations and responsibly sanitized where applicable to protect the identities of those involved.

By sharing these insights, we hope to normalize the discussion of vendor visibility and demonstrate how our unique perspective empowers security technology developers. The insights made available in this report have contributed to improving feature efficacy and overall safety within the Elastic Security solution. We hope security professionals everywhere can harness this report for their organization and the community at large.

Executive summary

One of the greatest challenges in security is providing universal access to the knowledge and resources necessary to protect against adversaries. Not every organization has a reverse engineer or intelligence analyst on the team, and this is especially true after macroeconomic conditions worsened earlier this year and resulted in waves of layoffs. Elastic Security Labs was founded to help address this knowledge gap by sharing threat and intelligence research along with tools and other software we use to impact threats.

To this effort, Elastic regularly publishes articles describing malware, threat intelligence, artificial intelligence, and detection engineering research along with tools and indicators. We have exposed more than a dozen previously unknown threats to the public and worked with our industry partners to develop new mitigations for them.

The 2023 Elastic Global Threat Report is a summary of more than a billion data points distilled down to a small number of distinct categories. We describe the tools, tactics, and procedures of threats from the perspective of endpoints and cloud infrastructure — the most common enterprise attack surfaces — so readers with varying priorities can determine the best course of action to take next.

As enterprises seize control of their environments, adversaries are pushed to the very edge of those spaces — congregating instead in network appliances, cloud infrastructure, and trusted third parties. Threat actors within the borders of an enterprise have been pushed to innovate and are researching ways to tamper with the sensors that make the environment so effectively hostile to them. *Defense Evasion* remains the single greatest adversary investment, cost, and challenge for those operating inside the enterprise, and it is becoming increasingly important at the edge.

Based on our observations from this report, we expect to see adversaries attacking security technologies by leveraging vulnerable device drivers and other methods to disable or circumvent controls. Risk averse adversaries may directly deal with cloud-resident data to minimize costs associated with being discovered. In uncovering several related espionage-motivated threat groups, we learned how much they depend on open sources for their capabilities. Organizations need to understand that the barriers to entry are gone — public projects published pseudo-anonymously are empowering threats of all kinds.

In spite of the greatest advances in security technology, global information sharing networks, and broad public awareness, the industry hasn't managed to solve security. Now, more than ever before, we're poised to achieve something meaningful — not just for the massive but for the masses.

Trends and correlations

The first section of the Elastic Global Threat Report contains trends reflecting the major tools, tactics, and procedures used by threat actors and observed in Elastic telemetry. Because each vendor has their own unique visibility, reports like this offer valuable insight into the capabilities used to monitor and disrupt threats. Elastic telemetry incorporates data from a variety of sensors, including Elastic Agent[1].

The following subsections contain insights into trends for malware signatures, endpoint behavior, and cloud security as observed during the prior year by Elastic.

Malware signature trends

Malware signatures play an important role in preventing malicious software from being executed or installed, and are supported on all major operating systems (OS). YARA signatures provide one layer of defense within the Elastic Security solution, identifying malware-related threat activity based on strings or byte-sequences. Elastic Security for Endpoint offers signatures at the file- and memory-level for all common endpoint operating systems and makes signatures available to the community through the protections artifacts repository as part of our free and open commitment.

Elastic Security Labs has analyzed specific operating system trends for malware according to our telemetry. From this, we identified that ~5.7% of all malware infections were on Windows endpoints, while ~91.2% were on Linux endpoints. Windows-based signature events dropped from 39.4% to about 5.7% of signature events, and infection rates similarly appear to be decreasing. Compared to 2022, these events appear to be down by about 85%. macOS signature events remain the smallest percentage.

> Linux-based signature events continued to increase from 54.5% last year to 91.2% of all signature telemetry. This likely has more to do with the overall adoption of Linux-based infrastructure than threat priorities, and may also influence our visibility of Linux-based malware infections — which we estimate have risen to about 59.8%.

1 The Elastic Security solution telemetry is generated by a diverse population of sensors and data sources which are too numerous to describe concisely, including sensors not developed by Elastic.

Malware by Category

Trojan	61.0%
Cryptominer	21.8%
Proxy	12.5%
Backdoor	1.4%
RemoteAdmin	1.2%
Ransomware	0.6%
Rootkit	0.5%
Shellcode	0.4%
Packer	0.3%
PUP	0.1%
AttackSimulation	0.1%

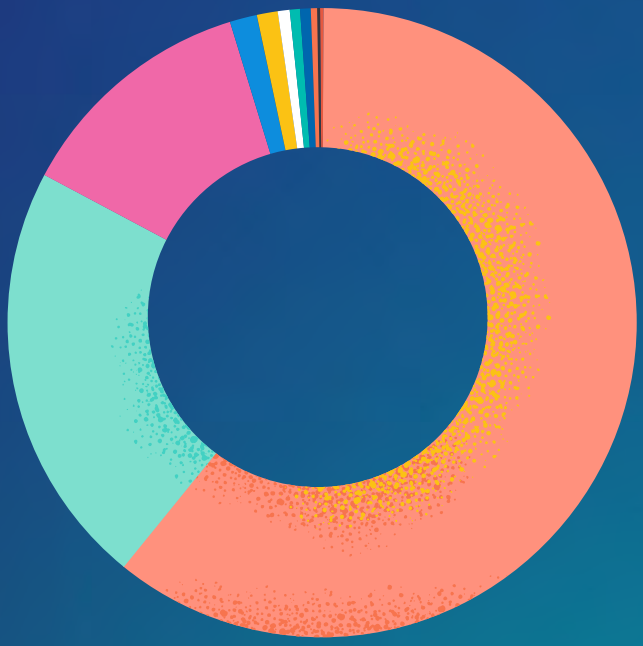


Figure 1: *Malware by Category*

Malware Infections by Operating System

Linux	92.1%
Windows	5.7%
Multiple	1.4%
macOS	0.9%

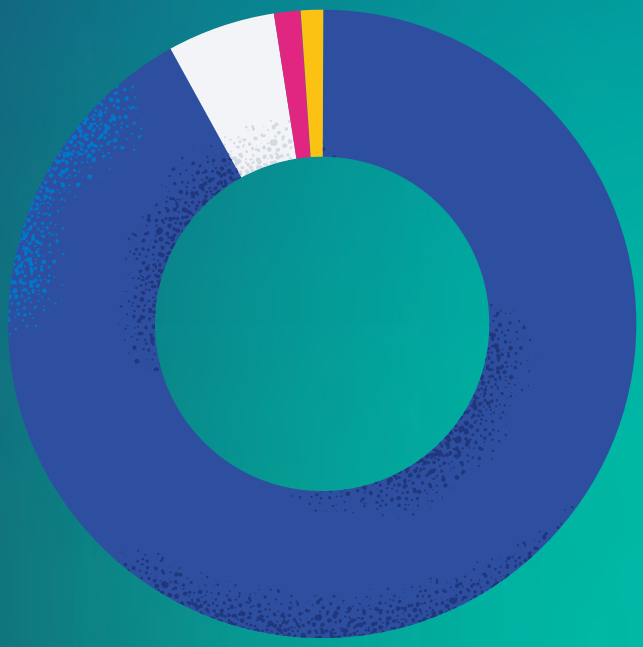


Figure 2: *Malware infections by OS*

During our analysis, Elastic observed 104 unique signatures. Figure 3 represents the top 10.

The majority of malware observed by Elastic Security Labs was composed of a small number of highly-prevalent families: Gafgyt, Frp, various ransomware-as-a-service (RaaS) tools, Meterpreter, and BlackCat. More than a third of all malware samples delivered to endpoints were associated with financially-motivated threats.

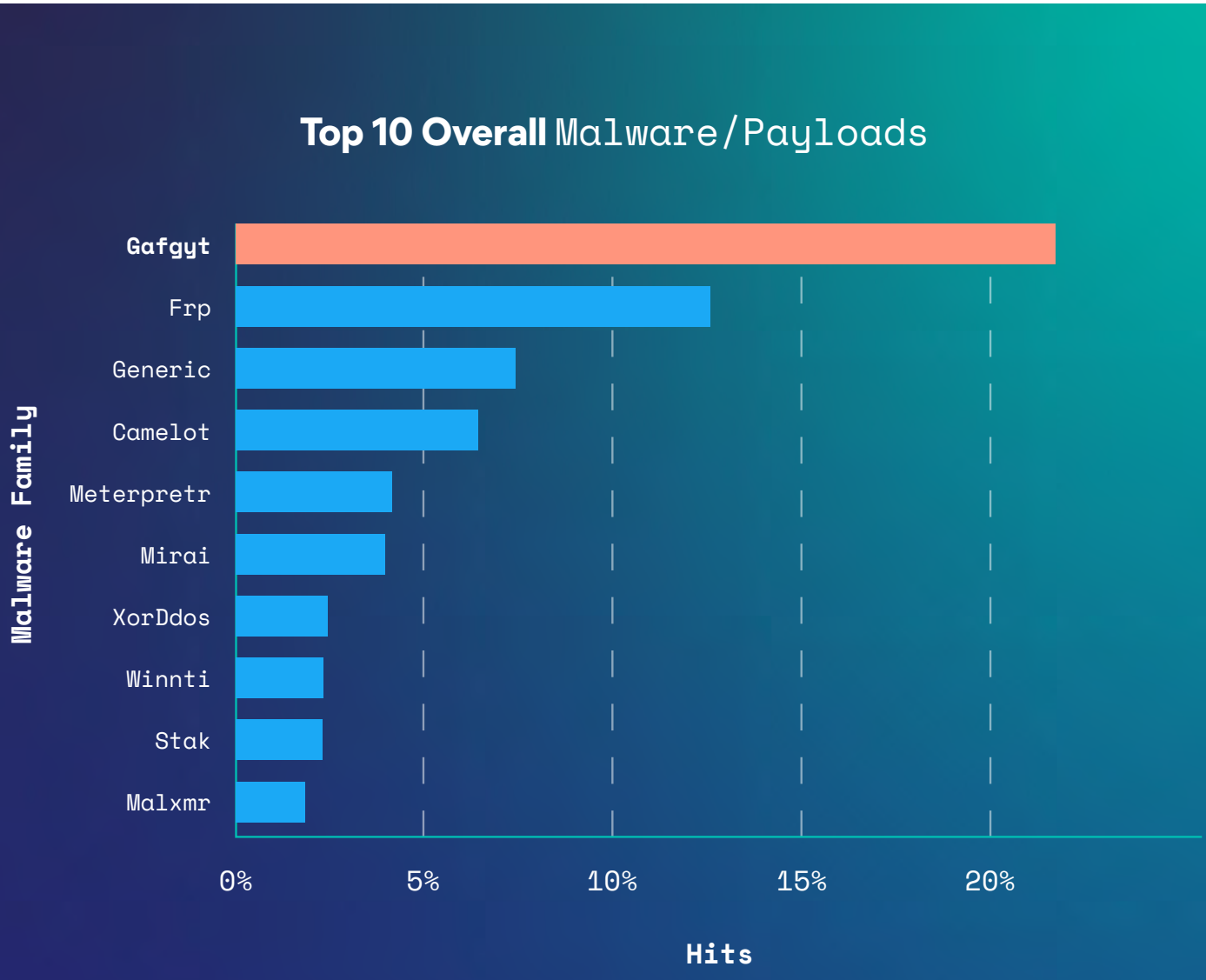


Figure 3: *Top 10 overall malware/payloads observed*

Like last year, we continue to see commercial off-the-shelf (COTS) capabilities like Mestaploit and Cobalt Strike employed by threat groups of all kinds. Sliver, a newcomer to the landscape, along with these commercial tools, represents about 6.8% of all implants we observed. Meterpreter, a key component of the Metasploit framework, is one of the few highly-prevalent malware families we see impact Windows, Linux, and macOS.

We also note the high degree of open-source capability adoption, with numerous sightings of Sliver, Donut, and SharpShares to highlight a few. In several espionage-related intrusions described by Elastic, researchers identified open-source projects co-opted by threat groups to achieve their goals. For example, Elastic saw tools like DNS-Persist, which was used to develop the **SOMNIRECORD** malware family.

This trend may support predictions of the continued rise of malware-as-a-service (MaaS) using COTS pen testing frameworks. Furthermore, the spike in ransomware also supports this trend, with a number of the most popular families being known for usage in MaaS/RaaS campaigns.

Sliver is similar to Cobalt Strike in terms of popularity as seen in raw infection numbers while being an open-source tool with arguably less polish and support. This may support the idea that threat actors are seeking to reduce the financial burden of their attacks. For example, it is common for the Conti ransomware to pair their infection with Cobalt Strike payloads for lateral movement and staging to deploy the ransomware. Being able to replace a COTS tool with an open-source tool such as Sliver reduces the cost to achieve a similar effect.

Top Ransomware Families

Ransomware families tend to cluster around the output of a specific group or set of malicious actors that utilize a distinct pattern of techniques, tactics, and procedures (TTPs). Identifying these families with specific names or codenames is important for tracking the evolution of ransomware families over time and for attribution purposes.

In order to prepare for ransomware, security teams need to familiarize themselves with the most active families. The top three ransomware families observed by Elastic Security Labs were Sodinokibi, Hive, and BlackCat, all of which are well-known families that possess capabilities which have been documented extensively by security researchers.

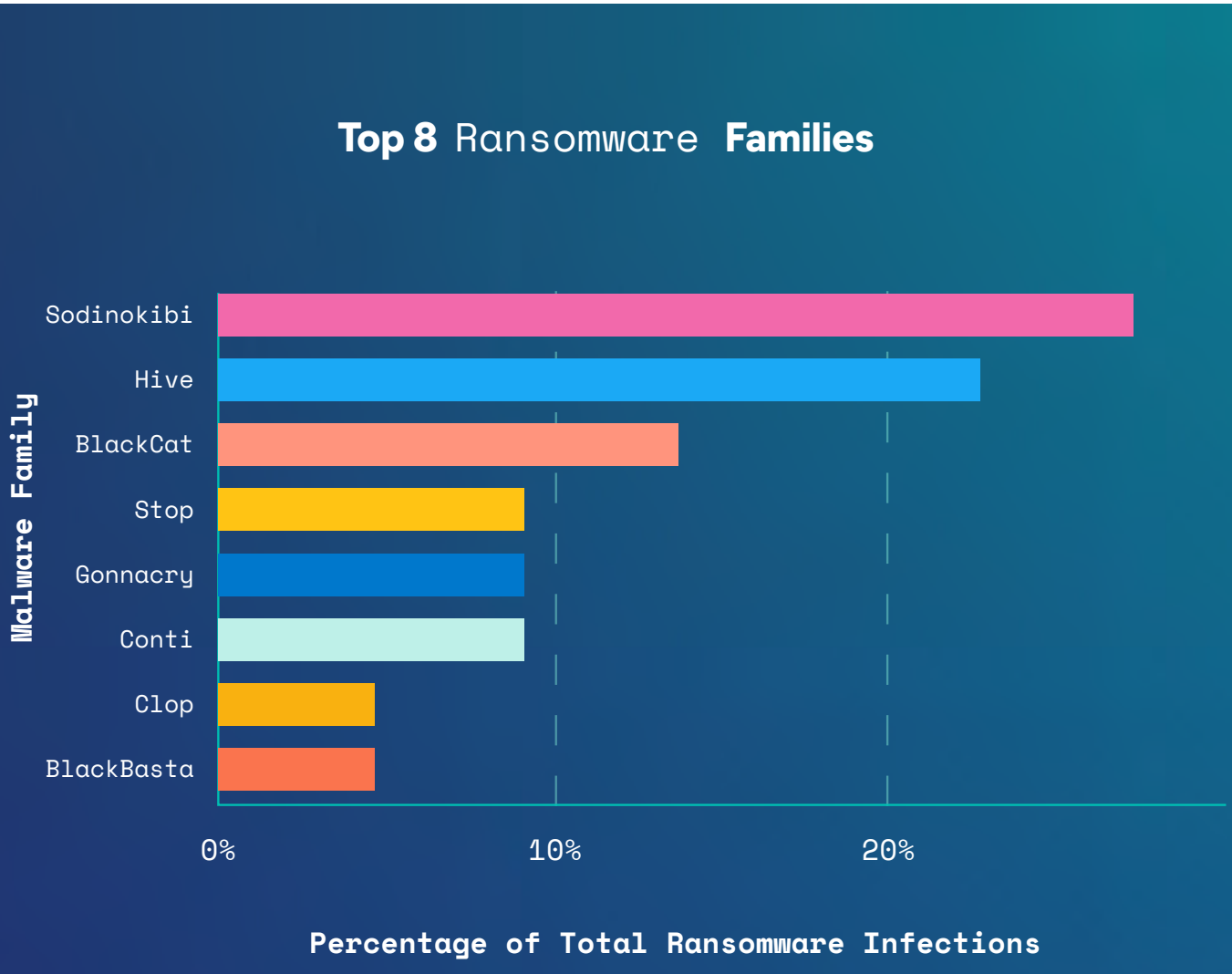


Figure 4: *Top 8 ransomware families observed*

Some background on Sodinokibi, Hive, and BlackCat

Sodikonibi, also known as REvil, was a very commonly seen ransomware family that purportedly had ties to GandCrab as well as DarkSide. The developers behind Sodikonibi chose to distribute their ransomware directly to malicious actors via the RaaS model.

Hive ransomware was also distributed via a RaaS model. The infrastructure utilized by Hive was notably **shut down** by the United States Federal Bureau of Investigation (FBI) in early 2023, which effectively disbanded the group behind the ransomware. Our telemetry matches up with the timeline of the FBI's efforts, as we did not see any further activity attributed to Hive after December 2022.

BlackCat, which leverages RaaS as well, has garnered attention due to its ties with former members of DarkSide, BlackMatter, and REvil. A distinctive trait of this group is their utilization of Rust programming language to craft applications that are both secure and efficient. Notably, BlackCat employs sophisticated attack methods such as double and triple extortion techniques.

The top three ransomware families we observed over the past year, visualized in figure 5, all leverage a RaaS distribution model. With the RaaS model, it is incumbent upon the actors to deploy the ransomware against their chosen targets; the developers are left to focus on building the ransomware and the necessary support infrastructure required to track their targets and ransom payments. This division of labor significantly lowers the barrier to entry for aspiring malicious actors that have the motivation but lack the requisite low level development skills to build their own ransomware. Developers collect upfront costs from their cybercriminal customers and are also assured a cut of future ransom payments collected by the actors operating on their behalf.

> Sodinokibi, Hive, and BlackCat have all distributed both Windows and Linux variants, which can likely be attributed to resource allocation made possible through the profitable RaaS model.

Ransomware families tend to evolve quickly after their initial discovery. As with any software, developers will resolve critical bugs or may implement features that didn't make the initial release. An example of ransomware evolution was observed last year when Hive switched codebases from the relatively flexible Go programming language to the more hardened and rigid Rust. In addition to migrating their codebase to a different language, implemented additional anti-reversing countermeasures and completely refactored their encryption algorithm implementation.

Top 10 Linux Malware/Payloads

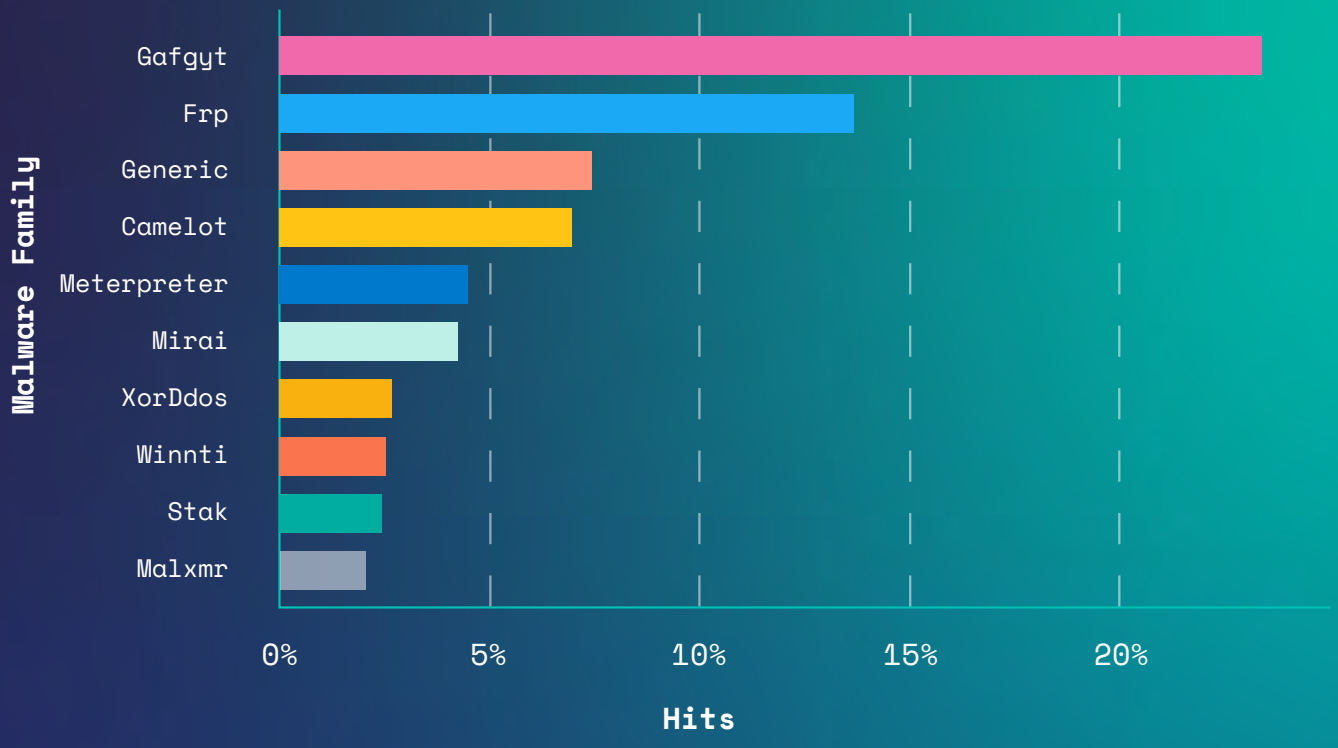


Figure 5: *Top 10 malware/payloads observed in Linux*

Percentage Linux Trojans

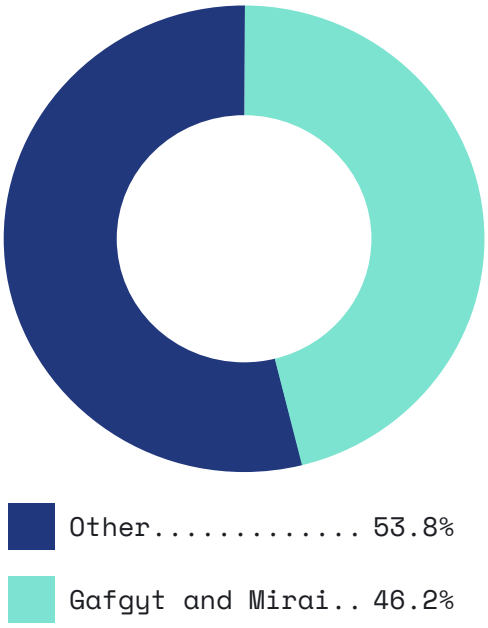


Figure 6: *A breakdown of the Trojans observed in Linux*

Elastic observed that cryptominers targeted Linux more aggressively than other operating systems. Despite this, botnet malware remained popular with Gafgyt, Frp, and XorDdos appearing most often. Cryptominers as a whole have seen a slight decrease in our observations but still remain popular, accounting for about 2% of detections. The popularity of cryptomining on Linux over Windows could be attributed to better support for cryptocurrency tools and libraries, as well as system stability and performance.

XorDdos is still prevalent and continues to rise in a similar pattern to what we saw in 2022, with a steady increase in the last 6 months of 2023, as shown below. The purpose of XorDdos is primarily to collect infected machines for use in denial of service botnet activities. However, XorDdos specifically has been a favored tool in compromising Internet of Things (IoT) and cloud based devices and also often includes a kernel rootkit component.

XorDdos acting as a tool to compromise IoT may also provide threat actors with root access to those devices, giving additional means for them to accomplish their objectives. Exposed docker servers are a known target of XorDdos. These make for good targets as they can often be noisy to network sensors with various push/pull operations, which can be used to help disguise botnet activity. However, the supplied rootkit could also provide means for supply chain based attacks building off of the initial distributed denial of service (DDoS) botnet access, poisoning the docker containers to grow the botnet or accomplish other objectives.

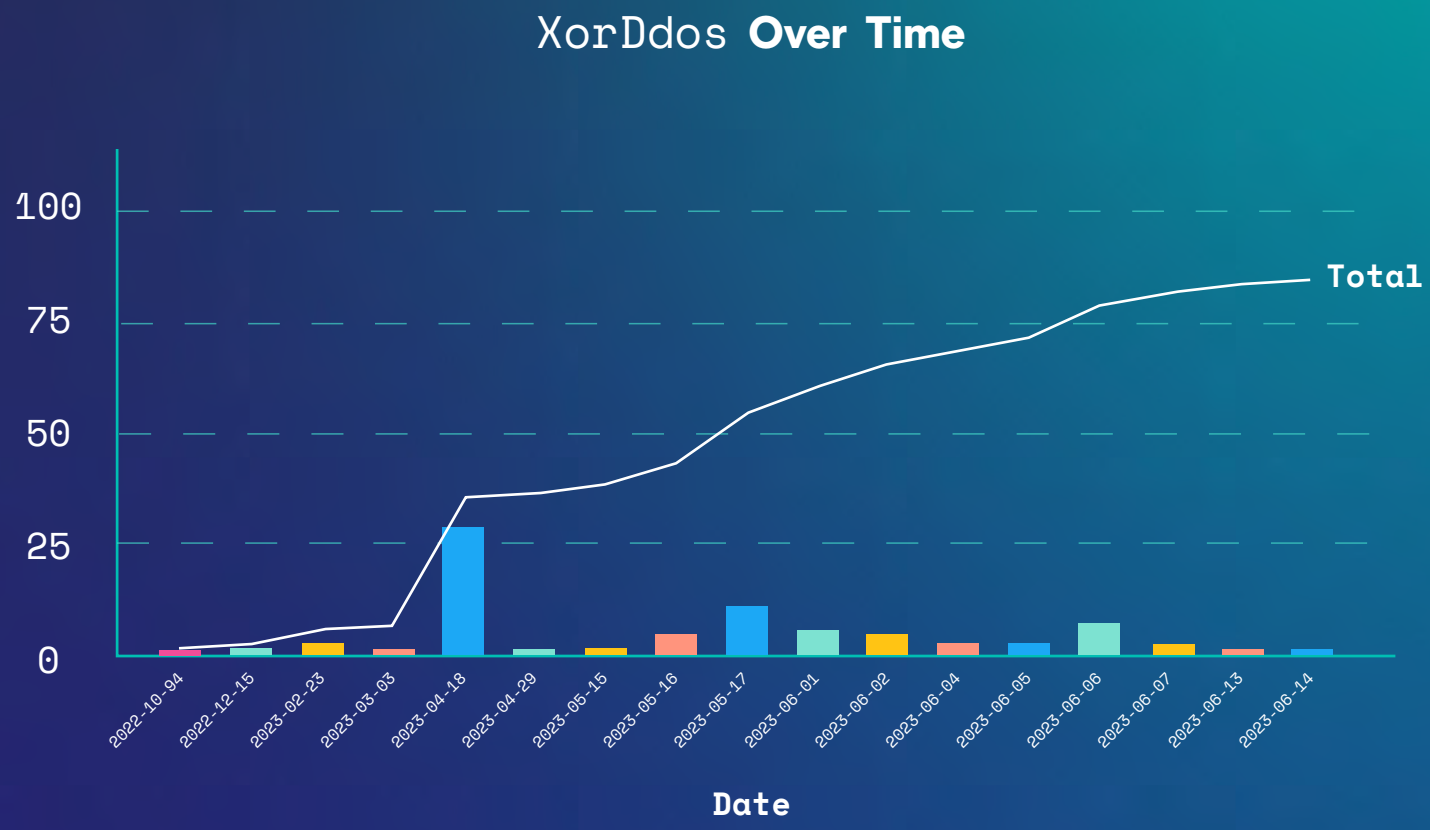


Figure 7: *The popularity of XorDdos from October 2022 to June 2023*

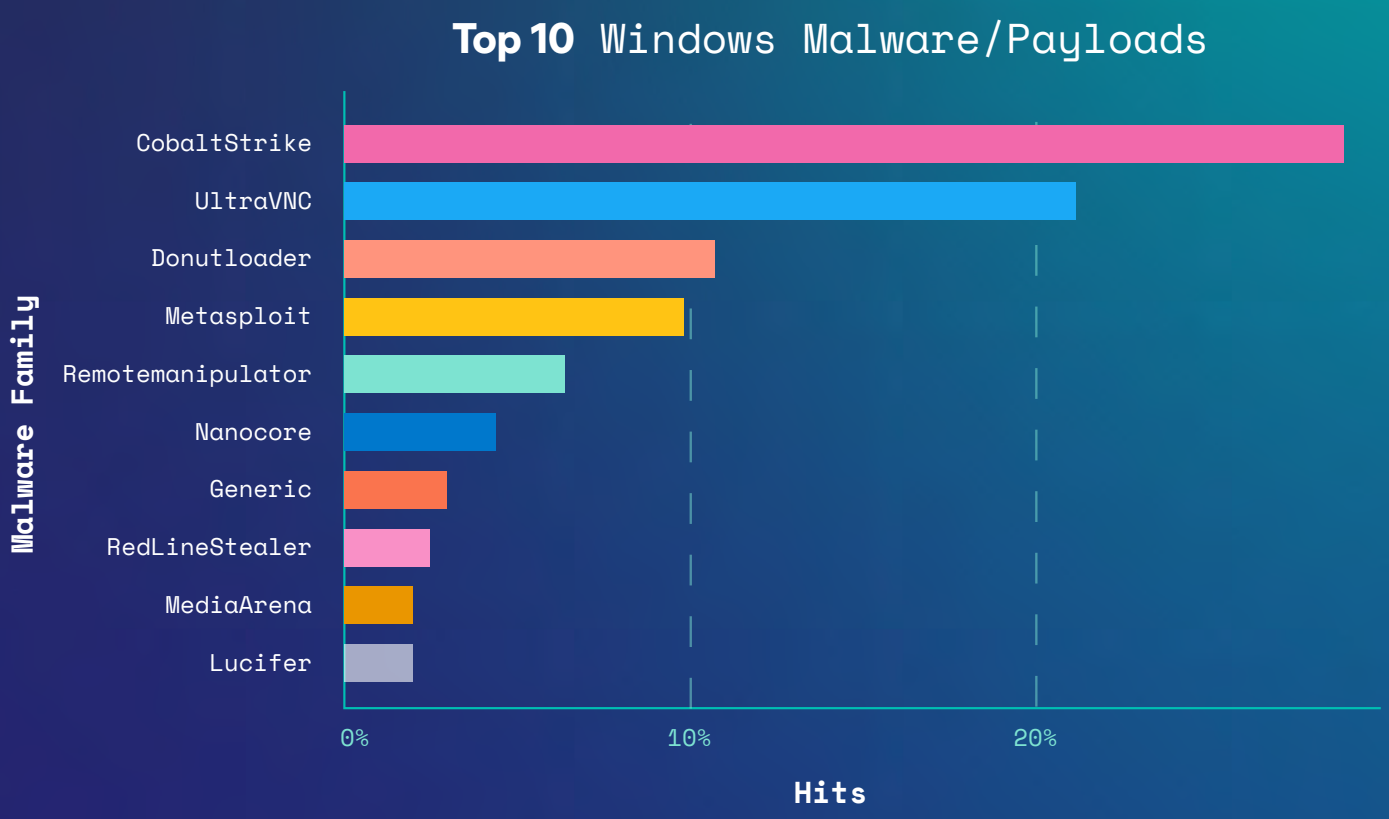


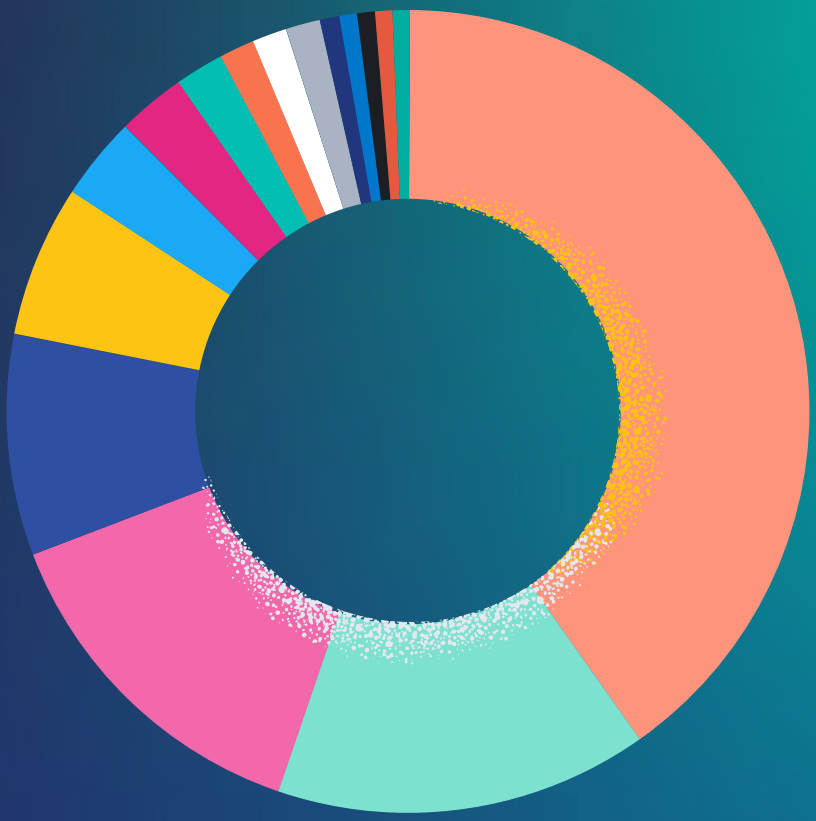
Figure 8: *Top 10 malware/payloads observed in Windows*

Top Malware in Windows

Windows-based signatures saw popularity from the malware families Metasploit and Cobalt Strike, which are used by threats of all kinds. Off-the-shelf capabilities enable less mature threats to improve their success rates, and amounted to about 68% of Windows infections. If we analyze malicious tools, those increased from about 6.5% last year to more than 21% this year.

This year, Nanocore overtook RedLineStealer in terms of Trojan popularity with nearly twice as many infections. Nanocore was first sighted in 2013 and is more of a classic Remote Access Trojan (RAT), while RedLineStealer is which is a more modern platform that has been on the scene for the past three years and offers an extensive set of features for would-be attackers. It is surprising to see Nanocore overtake RedLineStealer given the latter's modern capabilities. Despite that, both of these families represent only 8.8% of total malware signals for Windows Endpoints.

Trojan **Popularity for Windows Endpoint**



CobaltStrike.....	40.4%	SiestaGraph.....	1.4%
Donut.....	15.1%	DoorMe.....	1.4%
Mataspoit.....	13.7%	AgentTesla.....	1.4%
Remotemanipulator..	8.9%	ShadowPad.....	0.7%
Nanocore.....	6.2%	Matanbuchus.....	0.7%
RedLineStealer.....	3.4%	JesterStealer.....	0.7%
Lucifer.....	2.7%	IcedID.....	0.7%
Qbot.....	2.1%	Diceloader.....	0.7%

Figure 9: *Trojan popularity for Windows endpoints*

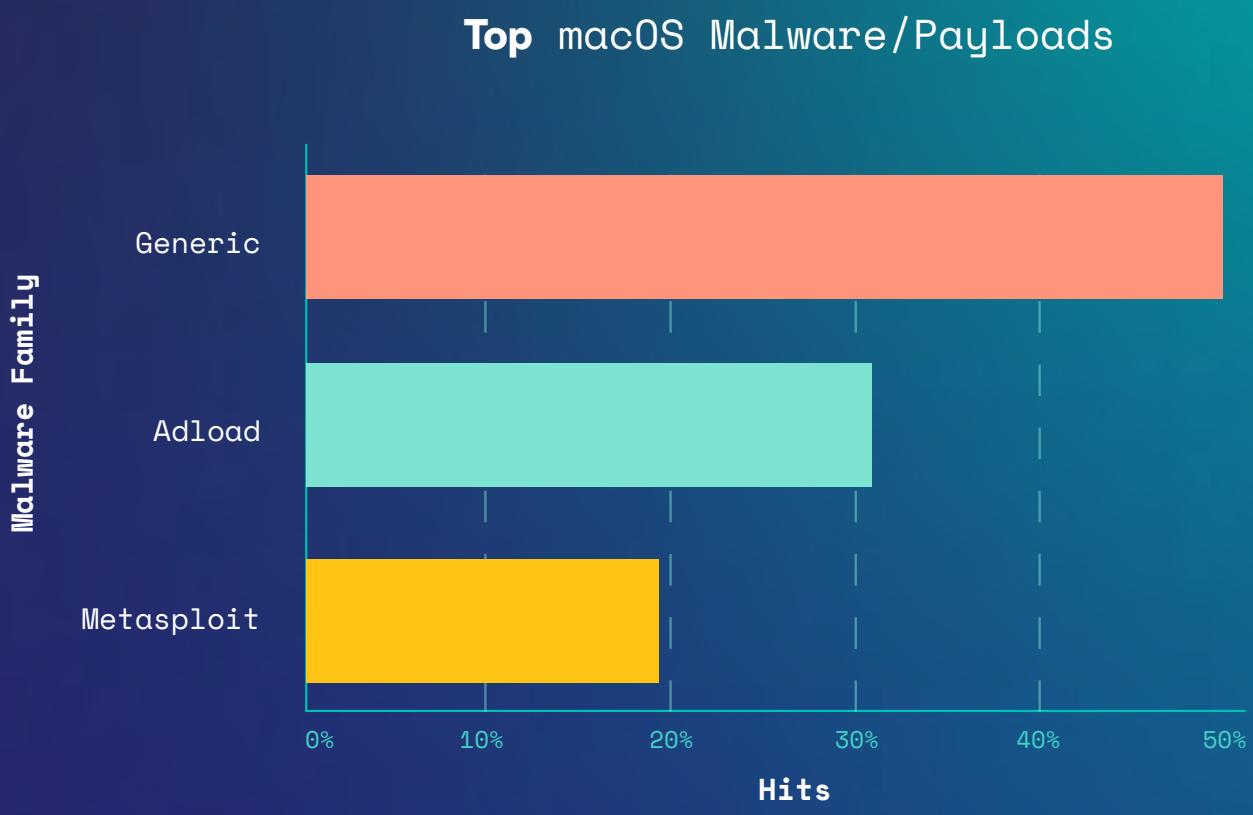


Figure 10: *Top malware/payloads observed on macOS*

Top malware in macOS

On macOS, the majority of malware infections were either general-purpose or commercially developed. This may be a function of our visibility and not a clear view of the macOS threat landscape. Metasploit, which was also a popular choice for targeting Windows and Linux endpoints, achieved the third place spot this year. Less prevalent but more powerful espionage-related capabilities like JOKERSPY demonstrated that macOS is not immune from malware.

Endpoint behavior trends

Malware is only one part of the story, and the capabilities present on enterprise endpoints can be dangerous if security teams are unaware of the ways that threat actors are operating them. In order to better protect the corporate network, we'll explore the global trends observed by Elastic Security Labs between July 2022 and June 2023. The majority of endpoint behaviors we observe occur on Windows, with macOS and Linux sharing equivalent minorities.

Elastic observed the most endpoint behavior alerts on Windows with ~94% followed by macOS and Linux each at ~3%. While Windows remains a significant focus for adversaries, Elastic saw a ~55% decrease in signals for Windows, while macOS signals increased nearly 118%. This sort of deviation has shown through the novel discoveries of Elastic Security Labs with publications describing macOS threats such as JOKERSPY and RUSTBUCKET.

> The most sophisticated threat groups are evading security by withdrawing to edge devices, appliances, and other platforms where visibility is nascent at best.

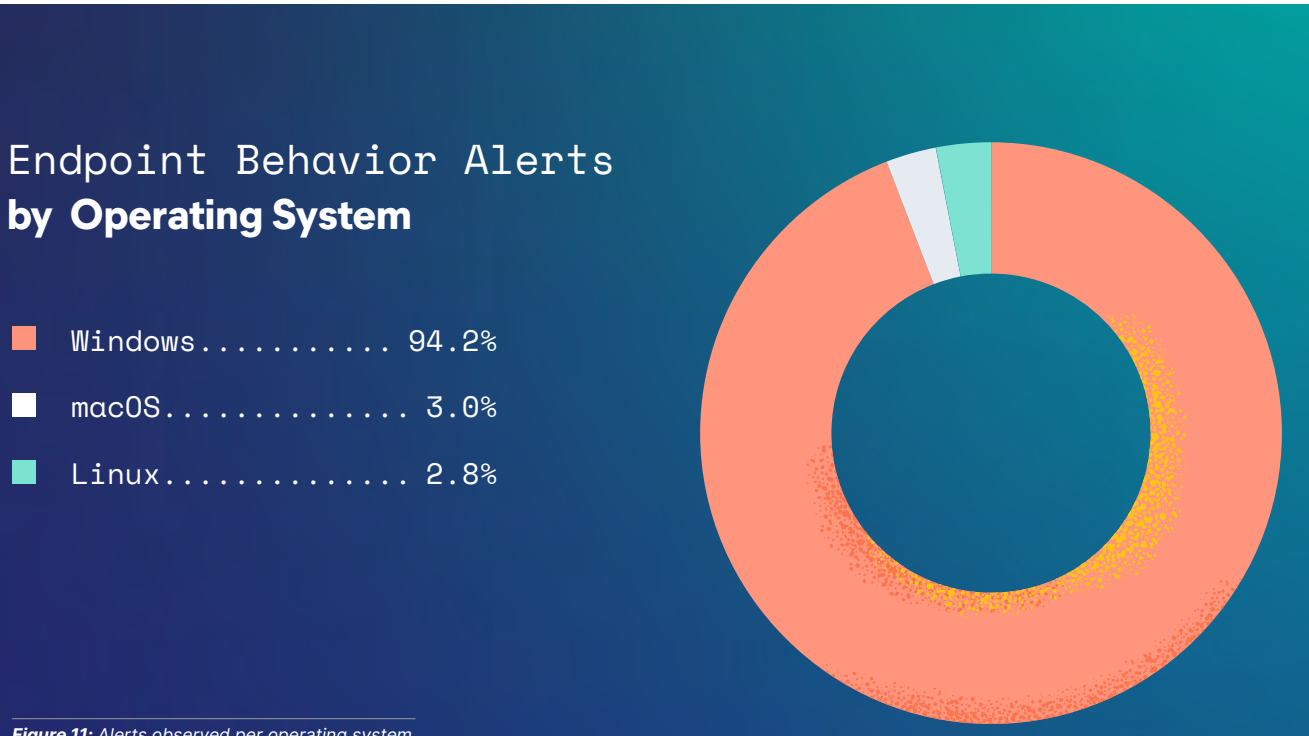


Figure 11: *Alerts observed per operating system*

Tactics	Hits
Defense Evasion	43.88%
Execution	29.20%
Persistence	7.98%
Privilege Escalation	6.93%
Credential Access	5.60%
Initial Access	2.82%
Lateral Movement	1.93%
Command and Control	1.13%
Discovery	0.31%

Table 1: *MITRE ATT&CK tactics observed across all endpoints*

Defense Evasion ranked the highest this year with 43% of all endpoint behavior alerts, followed by *Execution* at ~29% (consistent with last year). This indicates that adversaries are still focusing heavily on techniques designed to bypass detection. This can encompass activities such as disabling security software, obfuscating malicious scripts, or masquerading actions as benign processes. The high prevalence of *Defense Evasion* techniques suggests that attackers are well-aware of the monitoring tools and security solutions in place and are developing strategies to work around them. This is a clear sign that adversaries are adapting to hostile environments and that they're investing time and resources to ensure their malicious activities remain under the radar.

Execution at ~29% underscores the emphasis attackers place on running malicious code on compromised systems. Whether it's through malware, scripts, or leveraging legitimate system tools for malicious purposes, *Execution* is a fundamental stage in the cyber attack lifecycle. A high rate of *Execution* alerts could imply an increase in malware campaigns, or it might reflect the use of living off the land (LotL) techniques where attackers leverage built-in system tools to conduct malicious activities.

When looked at together, *Execution* and *Defense Evasion* make up more than 70% of all endpoint alerts, painting the picture of a typical attack pattern. After gaining initial access, attackers seek to run malicious payloads (*Execution*) and then immediately take steps to evade detection (*Defense Evasion*), allowing them to maintain a foothold in the compromised system.

The prevalence of *Persistence* at ~9% showcases the attacker's desire to maintain access to compromised systems over extended periods. Techniques in this category ensure that even if the initial malicious process is detected and terminated, other mechanisms will restart it, guaranteeing continued access to the target environment.

Lastly, the fact that Elastic Security Labs observed *Privilege Escalation* making up around 7% of total endpoint alerts is concerning. While this might seem like a smaller percentage compared to the others, it's significant given the potential impact. Once an attacker has escalated their privileges, they have a much broader range of actions they can perform on the compromised system, from accessing sensitive data to making changes at the kernel level. It indicates that attackers are not just content with gaining initial access; they're actively seeking to expand their control over compromised systems.

Defense Evasion

Masquerading and *System Binary Proxy Execution* accounted for ~69% of *Defense Evasion* techniques, but *Process Injection* still remains popular at ~13% of all endpoint behavior alerts. In Windows environments specifically, this may indicate that threat actors are focusing instead on native utilities rather than custom tooling.

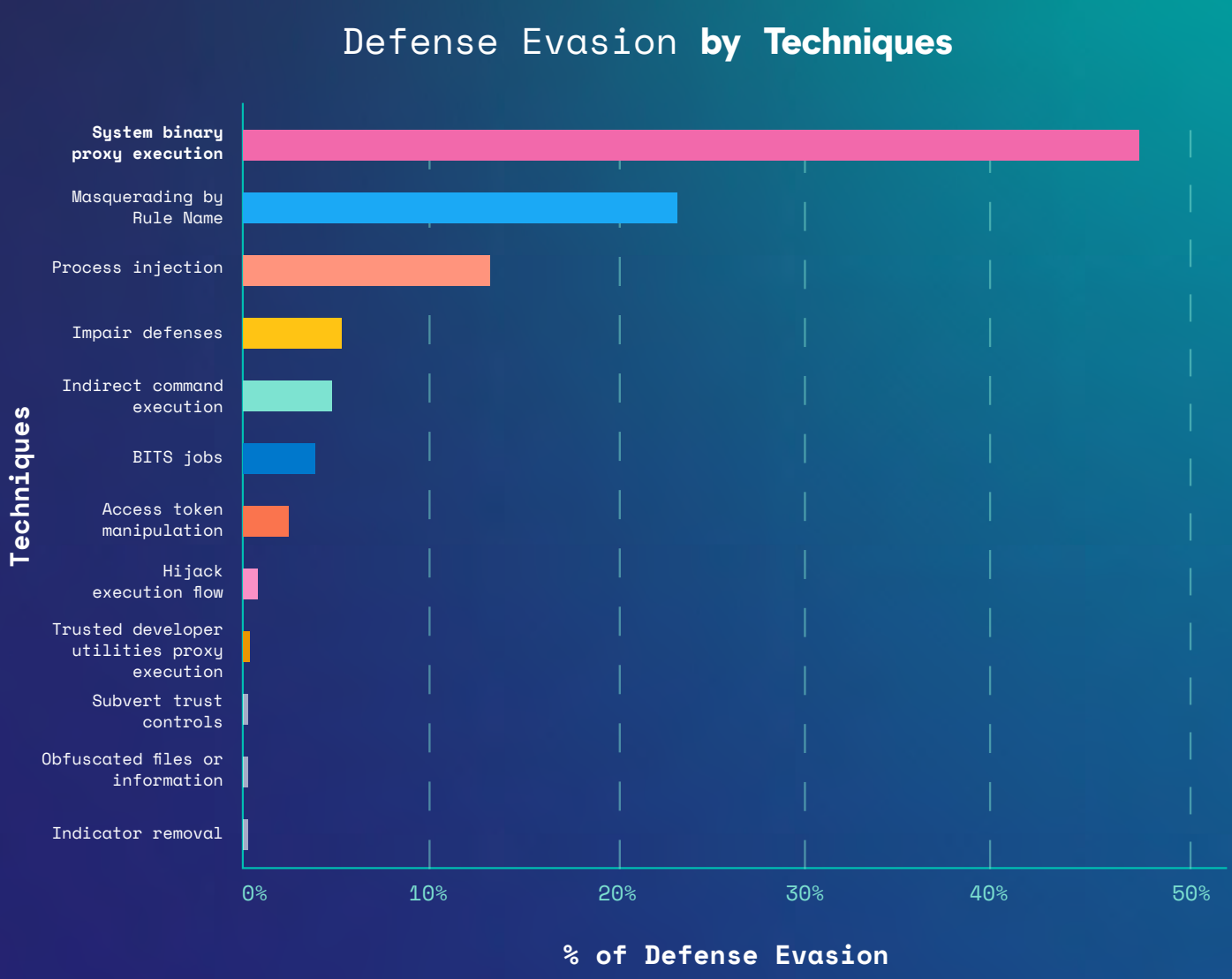


Figure 12: *A breakdown of Defense Evasion by techniques*

System Binary Proxy Execution was often favored by adversaries attempting *Defense Evasion* with ~48% of all *Defense Evasion* techniques. Often, this technique is coupled with the execution of malicious code depending on the sequence of events.

Unlike 2022, Elastic observed a notable amount of Windows Management Instrumentation (WMI) abuse, commonly detected by parent-child relationships where parent processes are either Microsoft Office applications or DLL and service creation processes. For example, to observe **winword.exe** being the effective parent process of **WmiPrvSE.exe** may indicate a malspam campaign in which a Microsoft Word document contained code to exploit ExquationEditor and call WMI for a payload execution. Another example would be using WMI to enumerate existing defenses as discussed by Elastic Security Labs' [QBot malware analysis research.]

System Binary Proxy Execution by Rule Name	Hits
Execution via a Suspicious WMI Client	23.95%
Script Execution via Microsoft HTML Application	18.76%
Execution of a Windows Script Downloaded via a LOLBIN	12.62%
Command Shell Activity Started via RunDLL32	7.93%
RunDLL32 with Unusual Arguments	6.91%
Execution of Commonly Abused Utilities via Explorer Trampoline	5.05%
Execution of a Windows Script File Written by a Suspicious Process	4.21%
Execution via Renamed Signed Binary Proxy	3.78%
Regsvr32 Scriptlet Execution	3.46%
Regsvr32 with Unusual Arguments	2.80%

Table 2: *A breakdown of System Binary Proxy Execution by Rule Name*

~31% of *System Binary Proxy Execution* occurred via the abuse of **rundll32.exe** and **mshta.exe** by HTML applications leveraging these utilities to execute malicious scripts. Elastic observed that 47% of *Defense Evasion* attempts abused **rundll32.exe** in some form or fashion. **rundll32.exe** continues to be a common and powerful utility for adversaries targeting Windows. As an example, a popular commodity malware, ICEDID, was observed leveraging **rundll32.exe** to call a **PluginInit** export as reported in the Thawing the Permafrost of ICEDID research paper from the Elastic Security Labs team.

Other common native utilities to execute malicious code and bypass security measures include **regsvr32.exe, msiexec.exe**, and **mshta.exe.** Leveraging native utilities like these presents a dual-faceted threat. Firstly, these are trusted Windows binaries, following the LotL Binaries (LOLBins) trend. Their legitimate status means they are less likely to be flagged by traditional security tools, allowing adversaries to bypass conventional detection mechanisms. Also, these utilities can execute or facilitate a variety of potentially malicious actions, from downloading and running scripts to registering malicious DLLs.

While **rundll32.exe** seems to reign supreme in terms of popularity for adversaries, **msiexec.exe** should not be ignored for DLL loading when performing threat hunting as highlighted by Elastic Security Labs in the research article Hunting for Suspicious Windows Libraries for Execution and Defense Evasion.

Next, we'll dive into the second highest *Defense Evasion* technique: *Masquerading by Rule Name*.

Masquerading by Rule Name	Hits
Binary Masquerading via Untrusted Paths	73.47%
Potential Masquerading as SVCHOST	10.85%
Renamed AutoIt Scripts Interpreter	6.05%
Evasion via Double File Extension	5.67%
Renamed Windows Automaton Script Interpreter	1.87%
Potential Binary Masquerading via Invalid Code Signature	1.34%
Renamed Third Party Administrator Tools	0.76%

Table 3: *A breakdown of observed Masquerading by Rule Name techniques*

~73% of *Masquerading* was accomplished by executing native Windows binaries or LOLBins in non-native native paths such as the **System32, Program Files**, and **SysWOW64** directories. Elastic's Binary Masquerading via Untrusted Path endpoint behavior rule has been vital to detecting and preventing such behavior as found with the SIESTAGRAPH and Operating Bleeding Bear research articles from Elastic Security Labs.

Figure 13 below showcases the most common binary names linked to *Binary Masquerading via Untrusted Paths* in a Windows environment. The top three — **rundll32.exe, svchost.exe, and MSBuild.exe** — are particularly noteworthy. Their widespread use in *Masquerading* underscores their trusted status within the OS. By mimicking these binaries, malicious activities can mimic genuine OS operations, potentially going undetected by conventional security tools.

> **rundll32.exe:** Historically, **rundll32.exe** is utilized to execute functions within DLL files, making it a favorite for adversaries due to its legitimate purpose of running code. An attacker can misuse this to execute malicious payloads, obfuscated within the appearance of a benign DLL execution.
>
> **svchost.exe:** This binary is responsible for hosting multiple Windows services. Since it often runs numerous instances simultaneously, adversaries can masquerade their malicious processes under its guise, blending in with the legitimate **svchost** processes and evading basic detection methods.

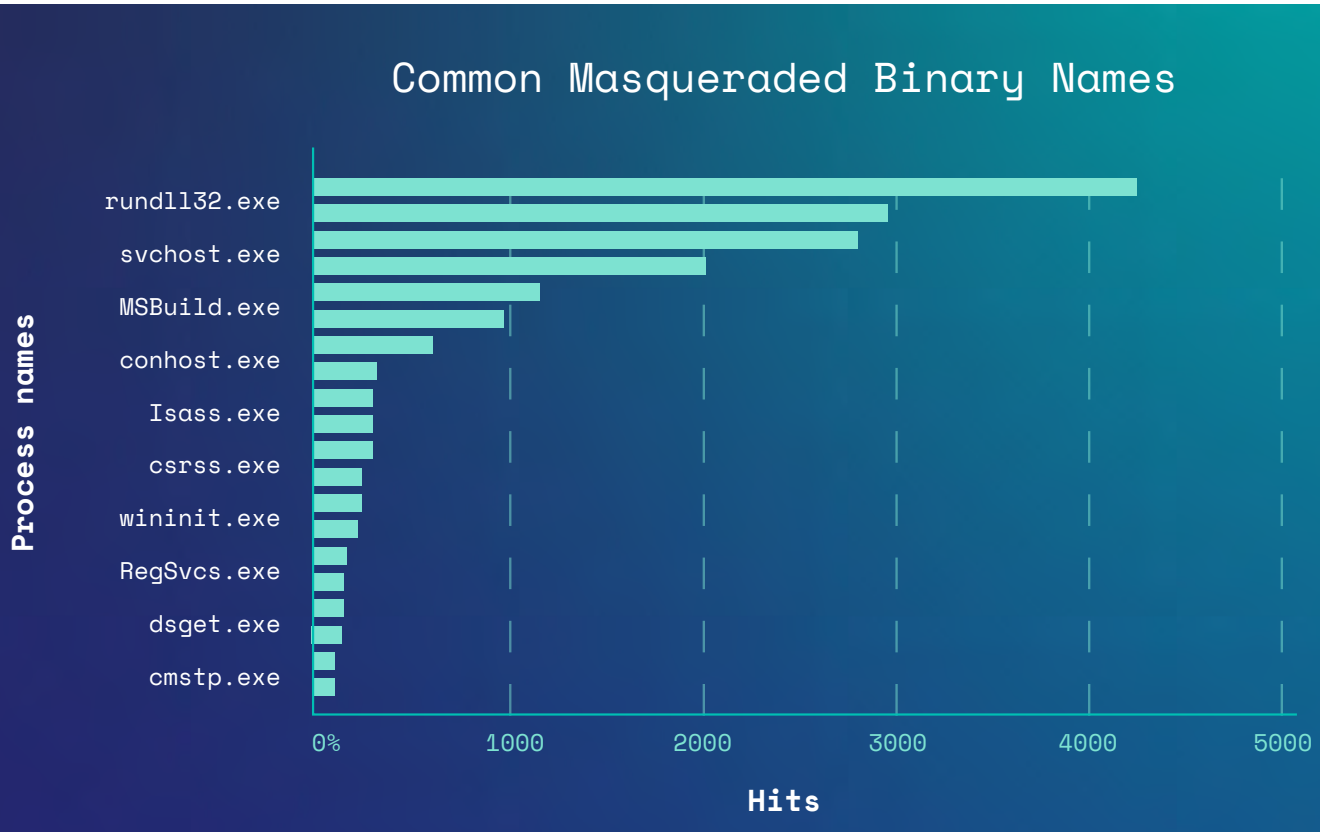


Figure 13: *Commonly masqueraded binary names*

Execution

The *Execution* tactic emphasizes methods that allow the adversary to run their own code in a target environment, like using a natively-available utility in Windows to execute a malicious JScript file. Within the *Execution* category, Elastic observed *Command and Scripting Interpreter* to account for an astounding ~76% of all *Execution* techniques, followed by *User Execution* with ~13%. Most operating systems support one or more scripting languages, in addition to any supported through third-party software.

Command and Scripting Interpreter being so common in adversary endpoint playbooks indicates a greater trend of adversaries leveraging built-in tools and scripting languages like PowerShell, Bash, or Python to execute their payloads. Living off the land, as we described in the overview section, allows attackers to blend into the environment and reduces the likelihood of detection since they're utilizing legitimate tools rather than custom malware. Living off the land techniques offer some important advantages to threat actors:

- **Stealth**: Using native tools can bypass traditional signature-based defenses because there's no malware binary to detect
- **Flexibility**: Scripting languages allow for a wide range of malicious activities, like gathering information or launching further attacks, all from a single script
- **Ease**: There's a plethora of scripts and tools readily available online, meaning attackers can easily modify existing scripts for their purposes rather than creating malware from scratch

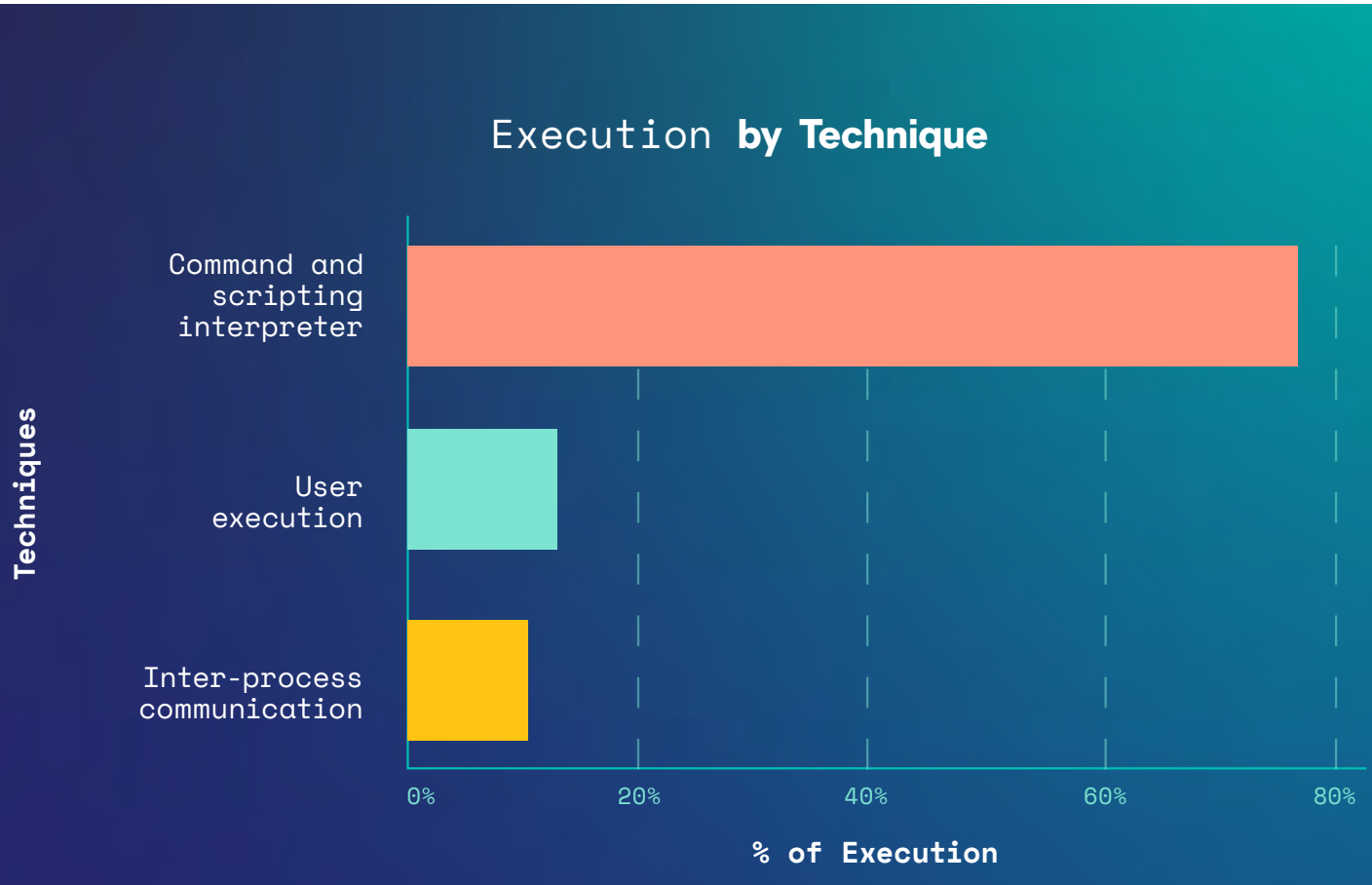


Figure 14: *A breakdown of Execution techniques*

For the *User Execution* technique, a distribution of 13% indicates that attackers still need to leverage capabilities that require user interaction, such as opening malicious documents or links. This also encompasses techniques like spear-phishing, where targets receive emails containing malicious attachments that execute a payload when opened. This proportion suggests a couple of things:

- **Human Factor:** Despite advancements in security technology, the human element remains a consistent vulnerability. Attackers understand that tricking a user into executing something can be more straightforward than bypassing technical defenses
- **Evolving Tactics:** While *User Execution* is a relatively traditional method, the high percentage means that attackers are still finding success with it, perhaps due to more sophisticated social engineering techniques or increasingly convincing malicious lures that reflect the increasing adoption of social and political media consumption. Elastic researchers have extensively shared effective hunting and detection techniques for *Execution and Defense Evasion* before.

Next, we'll return to *Command and Scripting Interpret* to look at some specific detections that Elastic observed.

Command and Scripting Interpret by Rule Name	Hits
Suspicious Windows Script Interpreter Child Process	24.53%
Suspicious Windows Command Shell Execution	14.99%
Inhibit System Recovery via Windows Command Shell	8.41%
Suspicious PowerShell Execution	7.01%
Execution from Unusual Directory	6.25%
Managed .NET Code Execution via PowerShell	6.13%
Execution of a Windows Script with Unusual File Extension	5.73%
Suspicious PowerShell Execution via Windows Scripts	5.02%
Windows Script Execution from Archive File	4.27%
Sudo Heap-Based Buffer Overflow Attempt	2.91%

Table 4: *A breakdown of Command and Scripting Interpret by rule name*

Based on observations from the table above, the most prevalent endpoint behavior within *Execution* was *Suspicious Windows Script Interpreter Child Process* with ~24%. This highlights a tendency from adversaries to manipulate Windows script interpreters for execution of malicious code. Additionally, the combination of *Suspicious Windows Command Shell Execution and Inhibit System Recovery via Windows Command Shell* point towards the usage of the Windows command shell in anomalous manners, including efforts to disrupt system recovery during execution efforts.

Various alerts related to PowerShell, namely *Suspicious PowerShell Execution and Managed .NET Code Execution via PowerShell*, underline its continuous popularity as not only a robust native Windows integration for administrators, but for adversaries as well. Elastic captured additional significant contextual information about these operations below.

- **Reliance on Windows Script Interpreters**. Adversaries regularly used binaries such as **wscript.exe, cscript.exe, mshta.exe, and powershell.exe.** These interpreters, inherent to the Windows environment, were chosen to execute scripts, circumventing the need for malicious binaries.
- **Use of script interpreters from anomalous locations**: Running native binaries such as **rundll32.exe**, **regsvr32.exe**, **and odbcconf.exe** strongly hinted at techniques like process hollowing, reflective DLL injection, or direct injection of malicious code payloads.
- **Diverse Script File Types for Malicious Activities As:** adversaries utilized a spectrum of script formats, including but not limited to HTA, VBS, JS, VBE, and JSE. The use of encoded formats, such as VBE and JSE, added an obfuscation layer, likely to impede script-based detection tools.
- **LOBSHOT malware**: Identified by our team earlier this year, leveraged **rundll32** to discreetly load and activate its malicious payload, demonstrating a cunning exploitation of legitimate Windows utilities to sidestep signature-based detection mechanisms.
- **Office Interactions with Windows Script Files (WSF)**: Trusted applications, namely Word, Excel, Outlook, and Equation Editor, were observed to unexpectedly drop, initiate, or modify WSFs. Such interactions likely indicated embedded malicious content, potentially harmful macros, or scripts designed for exploitation.
- **Use of Living-off-the-land Binaries (LOLBins):** We noted that legitimate utilities like **CertReq.exe, mshta.exe, rundll32.exe, and msiexec.exe** were exploited by adversaries. This co-option of trusted executables added layers of obfuscation, making detection and attribution more challenging.
- **Merging *Defense Evasion* with *Execution***: Telemetry indicated that adversaries combined techniques like *Masquerading* with the abuse of LOLBins or PowerShell. This fusion provided a two-fold advantage of concealing malicious activities while simultaneously facilitating execution.

Persistence

Persistence techniques are a top priority for adversaries post *Initial Access*. Their main goals are to maintain a foothold, execute malicious tasks, and remain undetected. These techniques serve as critical infrastructure for attackers, ensuring continuous malicious activities even through reboots or interruptions.

Scheduled tasks and jobs, especially in Windows environments, dominate the *Persistence* landscape, accounting for ~56% of techniques. This class of technique works by running adversary-controlled code (malware, typically) on a regular schedule. The substantial proportion demonstrates how adversaries exploit native features, like scheduled tasks, to blend in and decrease the likelihood of discovery and eviction.

Almost 17% of *Persistence* techniques observed involved *Boot or Logon Autostart Execution*, underlining its importance in an adversary's arsenal. At boot or user logon, these features execute adversary-controlled code (scripts, malware) to provide access. One of the most common sightings of this technique involved unusual child processes of **svchost.exe**, the service host application responsible for Windows task scheduling. Those descendant processes were launched from suspicious paths like the user's Public folder, **Windows\Tasks** and **\Windows\System32\Tasks** folders. These are commonly abused by adversaries to remain out of sight of the victim.

A staggering ~92% of prevented *Persistence* activities on hosts equipped with the Elastic Agent were detected through these suspicious child processes or irregular scheduled task creations. This frequency of occurrence underscores the necessity of vigilant monitoring, especially when focusing on child processes and scheduled tasks, two prime forms of *Masquerading* used by adversaries for *Persistence*.

Based on the current landscape, there's a growing reliance on script interpreters for persistent execution via Registry Run Keys and Startup Folder, which now make up approximately 17% of observed *Persistence* techniques. A good example of *Persistence* via **wscript** and scheduled tasks was identified in XWorm and AgentTesla campaigns described earlier this year by Elastic Security Labs.

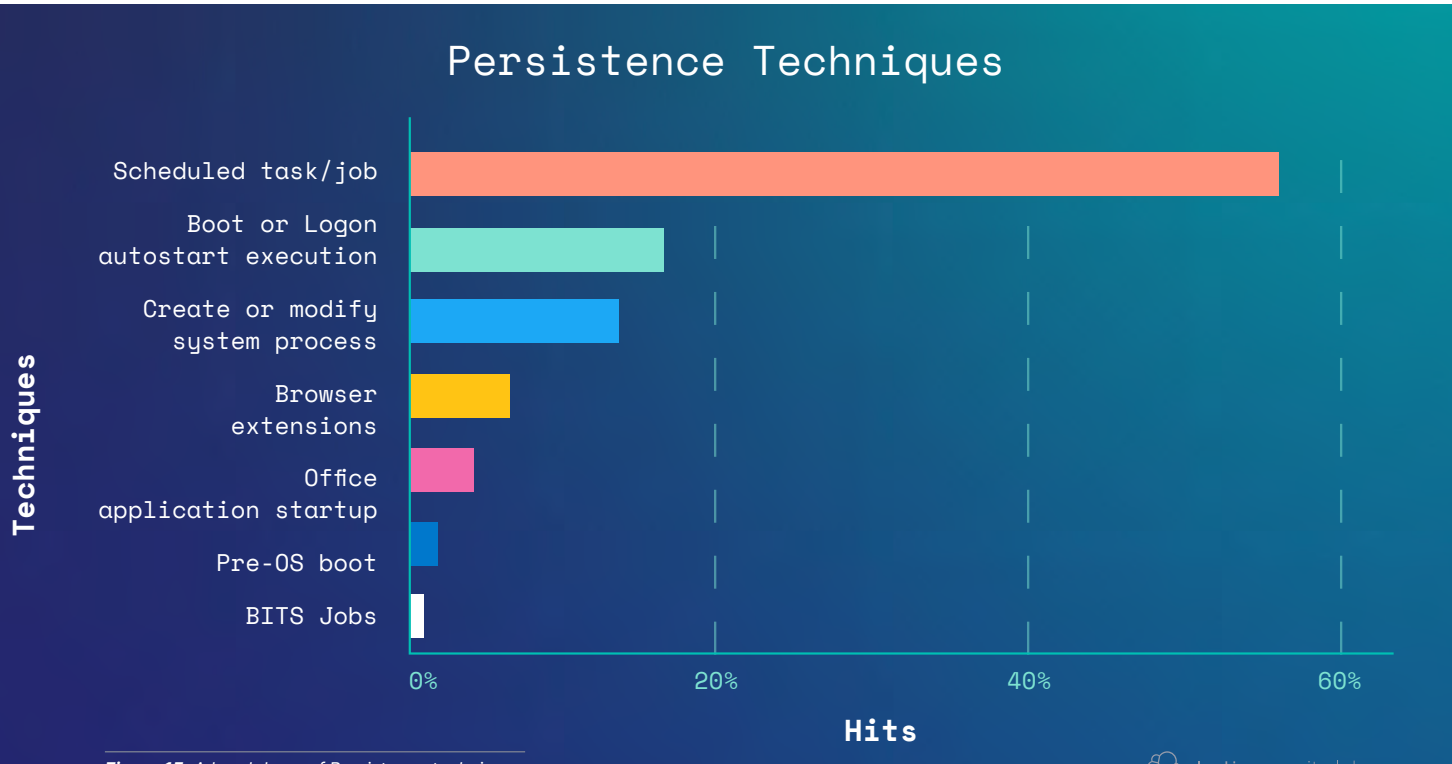


Figure 15: *A breakdown of Persistence techniques*

Privilege Escalation

Reduced to their core components, the techniques used for *Privilege Escalation* either directly or indirectly run commands or code in a higher privilege context. This can be used to infect a system, disable a security control, or modify the system configuration; typically *Privilege Escalation* is a means to an end and not an objective by itself.

Making up 27.50% of *Privilege Escalation* attempts, *Creating or Modifying System Processes* is a direct and often effective way for adversaries to both establish persistence and elevate privileges. This method typically involves adversaries spawning or altering critical system processes, such as **svchost.exe**, from non-standard locations or with unusual parent-child process relationships. By integrating themselves within legitimate-looking system processes, attackers can often bypass less sophisticated defense mechanisms. While this technique's popularity highlights its effectiveness, it also underscores the importance for defenders to implement process allowlisting and to keenly monitor any anomalies in process creation patterns — especially around system-critical executions.

Accounting for 20.31% of *Privilege Escalation* techniques, *Abusing Elevation Control Mechanisms* involves exploiting built-in Windows features intended for system management and security. Adversaries often target mechanisms like User Account Control (UAC), exploiting flaws or misconfigurations to gain elevated permissions. Techniques such as *UAC Bypass* employ various methods, including leveraging legitimate binaries (like **sdclt.exe**) known to have auto-elevated configurations without prompting the user. This technique's substantial prevalence reminds defenders of the double-edged sword nature of built-in security mechanisms — while they aim to protect, misconfigurations or overlooked vulnerabilities can ultimately become tools for attackers. As such, regular auditing of elevation controls and understanding of default OS behaviors become crucial.

Finally, *Access Token Manipulation* has emerged as the dominant technique this year by capturing 52.20% of observed *Privilege Escalation* attempts. The popularity stems from its ability to

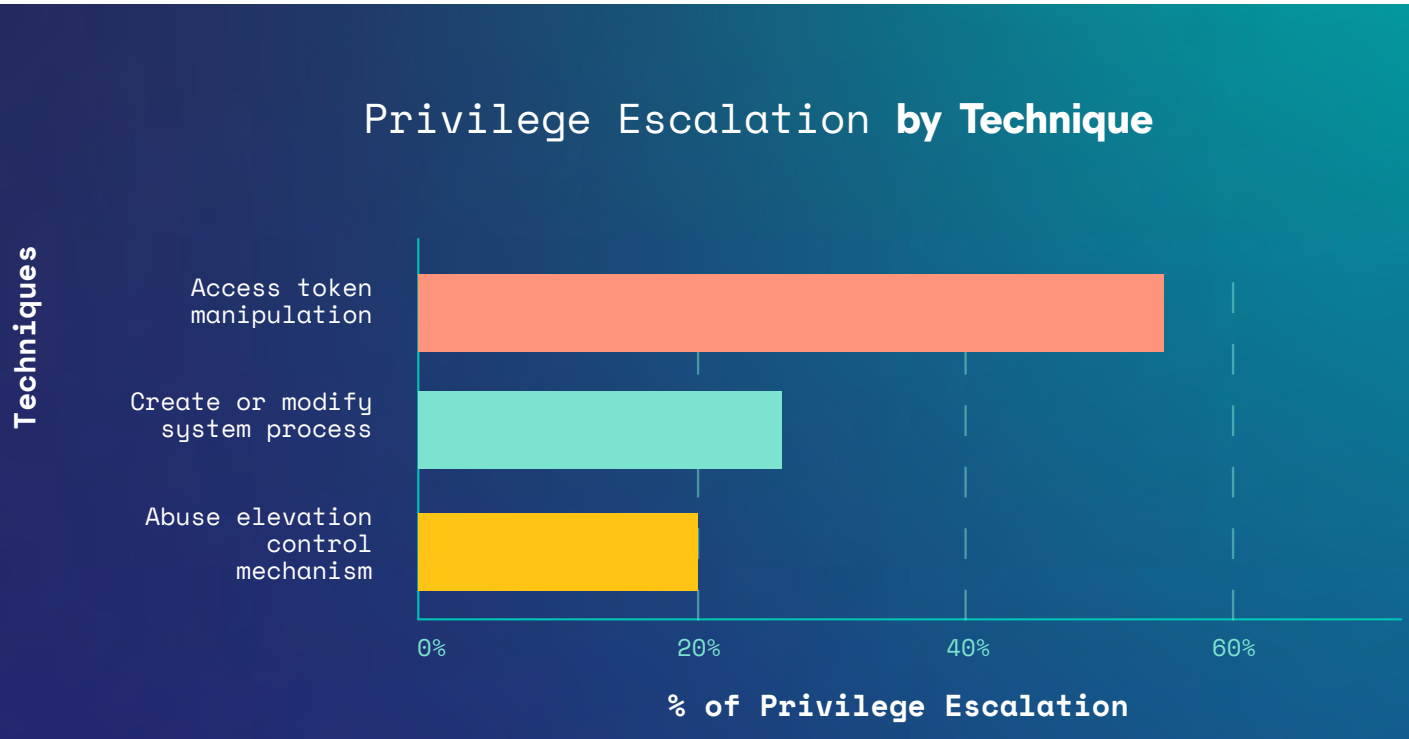


Figure 16: *A breakdown of Execution techniques*

stealthily seize and exploit user credentials and permissions. Adversaries leverage this technique by manipulating existing token privileges or stealing tokens from other processes to execute actions on behalf of another user, often leading to escalated privileges. Frequently targeted processes include those running with higher privileges, such as **lsass.exe**.

The pervasive use of this technique signals a challenge for defenders; continuous monitoring and validation of token integrity across system processes become imperative. The elevated percentage also indicates adversaries' preference for discreet lateral movement within a compromised environment. An example of this is SPECTRALVIPER, a novel finding published by Elastic Security Labs this year, which used access token impersonation.

We've broken *Access Token Manipulation* down even father into the rules detected by Elastic:

Access Token Manipulation by Rule Name	Hits
Privilege Escalation via EXTENDED STARTUPINFO	49.51%
Potential Privilege Escalation via Token Impersonation	38.18%
Access Token Manipulation via Child Process	6.03%
Unusual Privilege Escalation to System	3.14%
Privilege Escalation via Named Pipe Impersonation	3.14%

Table 5: *Access Token Manipulation by Rule Name*

Elastic observed adversaries exploiting the **EXTENDED STARTUPINFO** feature in Windows to enable *Privilege Escalation*. This occurred often in the wild, amounting to 49.51% of observed activities. This specific technique manipulates startup configurations to spawn processes with configurable security contexts.

> *When hunting for artifacts related to Access Token Manipulation, teams can:*
>
> - *Search for API calls invoking the* **CreateProcess** *function where the lpStartupInfo parameter points to an* **EXTENDED_STARTUPINFO_PRESENT** *structure.*
> - *Keep an eye out for abnormal relationships between parent and child processes, particularly those involving the* **lsass.exe** *process.*
> - *Detect certain paths such as* **\\Windows\\Temp*.exe**, *and invoking processes like* **msdtc.exe** *and* **taskhostw.exe** *from atypical locations.*
> - *Identify the creation of processes with Windows SID S-1-5-18 where a parent executable exists but the parent process ID is greater than* **0**, *indicating potential parent process spoofing.*

Elastic documented significant activities around *Token Impersonation*, a technique that accounted for 38.18% of detected *Privilege Escalation* attempts. *Token Impersonation* involves the theft or manipulation of process tokens to execute new processes with another user's credentials. Below are additional details Elastic Security Labs uses to detect such activity:

- Processes initiating under the **SYSTEM** user account
- Discrepancies like the mismatch between the real parent process and the one reported by the operating system. Notably, this involves processes like powershell.exe and paths such as **?:\\Windows\\system32*.exe** and **?:\\Windows\\SysWOW64*.exe.**
- Other indicative paths include **?:\\Windows\\Microsoft.Net*.exe, ?:\\Windows\\servicing\\TrustedInstaller.exe, ?:\\Program Files\\Microsoft*.exe**, and **?:\\Program Files (x86)\\Microsoft*.exe.**

Abuse Elevation Control Mechanism by Rule Name	Hits
Elevated Execution via ShellExecute RunAs Administrator	48.95%
UAC Bypass Attempt via AutoElevated Program Hijack	34.28%
UAC Bypass via FodHelper Execution Hijack	4.88%
UAC Bypass via DiskCleanup Scheduled Task Hijack	4.57%
UAC Bypass via ICMLuaUtil Elevated COM Interface	4.17%
UAC Bypass via Event Viewer	3.14%

Table 6: *Abuse Elevation Control Mechanism by Rule Name*

Although *Create or Modify System Process* represented ~28% of all endpoint *Privilege Escalation* techniques, Elastic opted to delve deeper into *Abuse Elevation Control Mechanism* activity and provide contextual information about observations from alert telemetry. Such mechanisms, like UAC in Windows are intended to segregate standard user functionalities from those requiring elevated privileges, thereby acting as a barrier against unintended or malicious changes. Significance in understanding this abuse lies in the fact that when successfully exploited, adversaries can execute their malicious payloads with heightened permissions, bypassing security measures. Often imbued in defensive layers, malicious payload execution can be blocked by the identification of the malicious action itself. If that fails, UAC is another layer defenders can rely on unless bypassed, where it becomes obsolete in preventing these actions. Below are explanations of the top two *Abuse Elevation Control Mechanisms* we detected:

- **48.95% – *Elevated Execution via ShellExecute RunAs Administrator*:** By using the **ShellExecute** function with the **RunAs** verb, attackers effectively bypassed UAC, elevating malicious scripts or binaries without explicit user consent. Given its widespread use, vigilance regarding such activity is paramount. Elastic researchers identified similar techniques with abusing **ShellExecute** during our research on popular commodity malware, Formbook, while also diving deep into how Elastic writes detection logic for UAC bypasses.
- **34.28% – *UAC Bypass Attempt via AutoElevated Program Hijack*:** Adversaries exploited Windows applications that are auto-elevated by design, enabling the execution of malicious payloads with heightened privileges. Such hijacking of auto-elevated binaries highlights the need for stringent monitoring of alterations or unusual activity involving them.

Credential Access

Adversaries of all kinds favor *Credential Access* techniques because they provide some of the most impactful utility. Credentials, which is a broad category encompassing everything from passwords to authentication tokens, open doors to data or systems. Nearly 7% of all endpoint behavior signals we observed related to this tactic category, and about 79% of those events involved OS-specific credential dumping techniques. Those are techniques that can be executed with tools or features built into various operating systems.

About 17% of the signals in this category involved input capture, primarily on macOS where **OSASCRIPT** was used to prompt users for credentials, and credentials from local password storage. In this section, we'll dissect both native and non-native *Credential Access* capabilities.

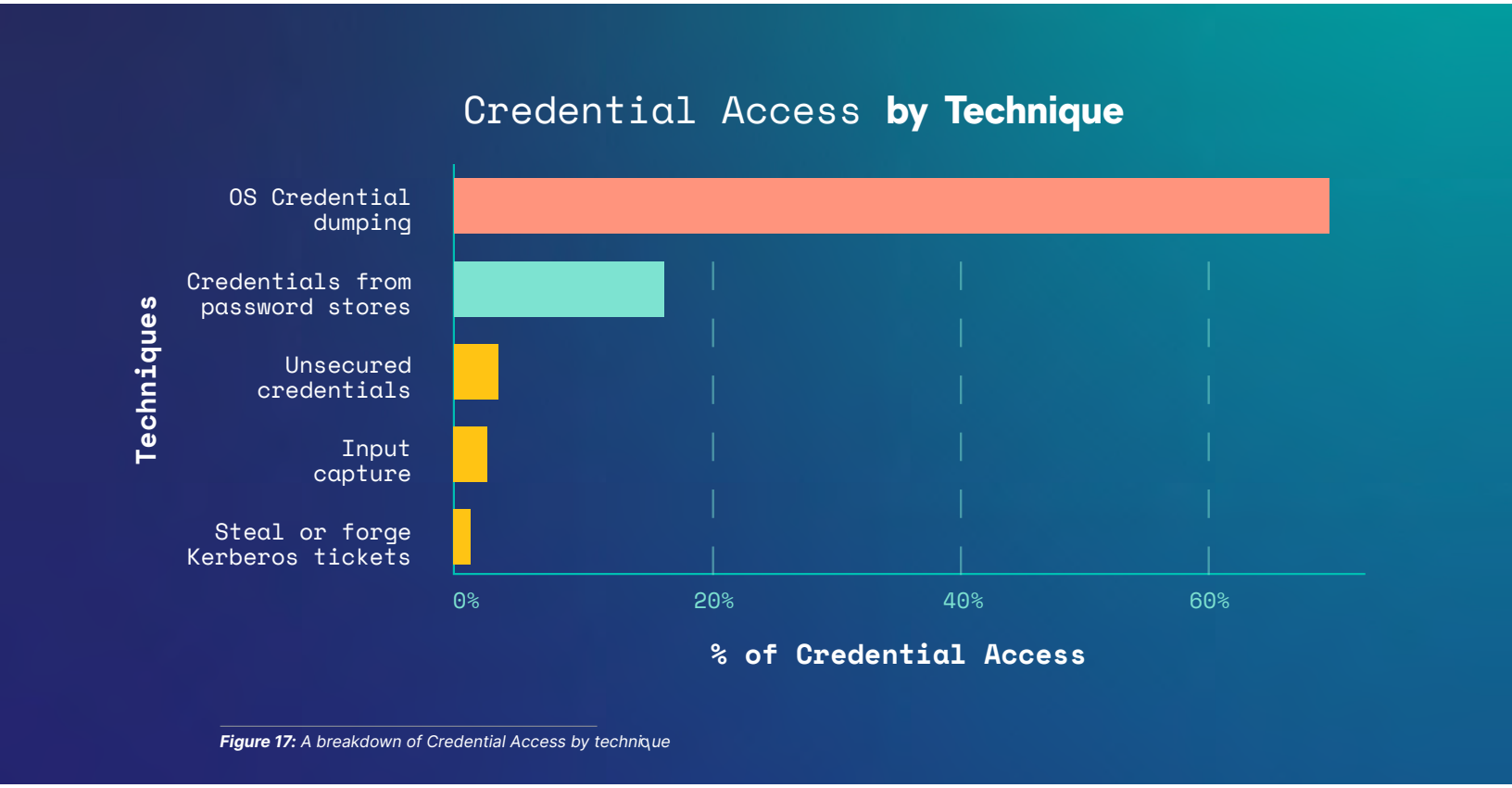


Figure 17: *A breakdown of Credential Access by technique*

Known utilities, which includes native tools like **reg.exe** and third-party tools like **LSASecretsDump.exe**, represented almost 80% of all *OS Credential Dumping* behaviors.

OS Credential Dumping by Rule Name	Hits
Credential Access via Known Utilities	79.81%
Potential Credential Access via Mimikatz	20.19%

Table 7: *OS Credential Dumping by Rule Name*

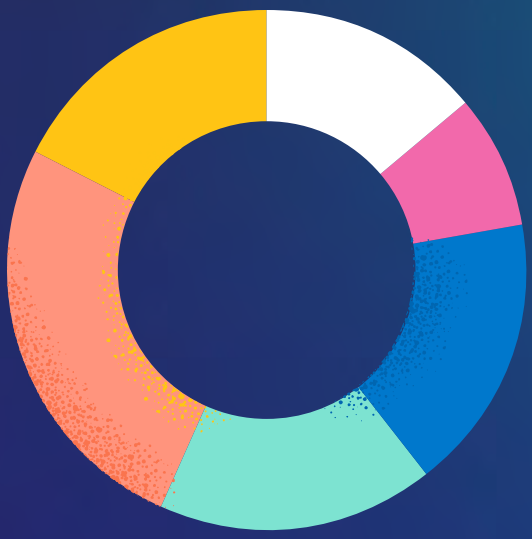
Elastic has identified several native Windows binaries that are frequently leveraged for *Credential Dumping* activities. At the forefront, **reg.exe** tops the list, especially when used with the **save** command-line argument. This tactic enables adversaries to save the Security Account Manager (SAM) database, a critical component of Windows that stores user account credentials in a hashed format. It's worth mentioning that **reg.exe** is not only used for *Credential Dumping*, but also helps adversaries create exclusions for defensive tools as detailed by Elastic Security Labs' QBot attack pattern analysis.

An unexpected finding was the frequent misuse of **netsh.exe** for malicious purposes. Adversaries leveraging this binary have been seen attempting to extract passwords related to wireless LAN (WLAN) configurations. By pairing **netsh.exe** with the arguments **show profile** and **key=clear**, and when executed with administrative privileges, adversaries can conveniently expose the plaintext password of any saved wireless network profile on the host.

> Other binaries like ***procdump.exe, rundll32.exe, and diskshadow.exe*** were often spotted in the context of *Credential Dumping*. ***Procdump.exe*** can be misused to dump process memory, which can then be parsed for credentials. ***Rundll32.exe*** allows attackers to execute code within the context of the Windows shared app processes, potentially accessing in-memory credentials. Meanwhile, ***diskshadow.exe*** can be exploited to bypass file locks, granting access to locked files like the ***NTDS.dit*** database — another rich source of credentials.

Non-native utilities remain a prevalent method of obtaining credentials. Mimikatz is one well-known example familiar to many due to the presence of immediately-recognizable arguments like **sekurlsa** and **lsadump**. A recent Elastic Security Labs article described methods of *Credential Access* threat hunting. Mimikatz made up a little more than 20%.

This year, we began monitoring how adversaries were targeting Password Stores. Unauthorized access to the Windows Password Vault not only jeopardizes individual credentials but can also provide attackers with the necessary privileges to further their attacks, move laterally within networks, or exfiltrate sensitive data. PowerShell, a built-in framework integrated with Windows, was used 62.85% of the time.



Common Native Utilities **for Credential Dumping**

- reg.exe..... 25.7%
- netsh.exe..... 17.4%
- diskshadow.exe.... 17.3%
- rundll32.exe..... 17.2%
- Other..... 14.0%
- procdump.exe..... 8.4%

Figure 18: *Common Native Utilities for Credential Dumping*

Credentials from Password Stores by Rule Name	Hits
Access to Windows Passwords Vault via Powershell	62.85%
Suspicious Vault Client Image Load	27.98%
Web Browsers Password Access via Command Line	9.16%

Table 8: *Credentials from Passwords Stores by Rule Name*

We have included a command-line example of how this is typically accomplished:

```
C:\WINDOWS\system32 windowspowershell\v1.0\powershell.exe
"[Windows.Security.Credentials.PasswordVault,Windows.Security.Credentia
ls,ContentType=WindowsRuntime];(new-object Windows.Security.Credentials.
PasswordVault).RetrieveAll() | % { $_.RetrievePassword(); $_ }"
```

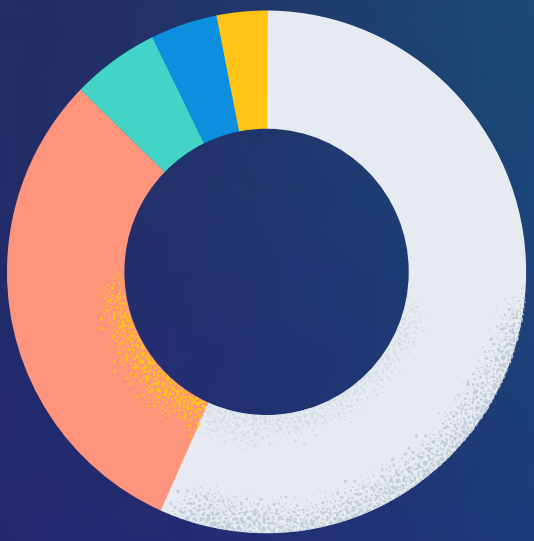
This command showcases how adversaries use PowerShell to instantiate the PasswordVault class and retrieves all of the credentials stored in the Credential Locker, including both usernames and passwords.

Cloud security trends

Securing cloud computing systems is a crucial part of cybersecurity. In order to protect these systems, cloud providers and users must prepare for the most likely changes to this attack surface. To identify these changes, Elastic utilizes global telemetry from customers who have agreed to share data about the pre-packaged detection rules they've enabled to analyze cloud-based threats and potential attacks.

This telemetry gives Elastic Security Labs tremendous insight into the potential threats customers see daily within Microsoft Azure, Amazon Web Services (AWS), and Google Cloud. The telemetry analyzed for this section is derived from the detection rule matches within Elastic's SIEM, which is why they're referred to as signals rather than attacks.

To understand our cloud-specific visibility, it's important to look at where our data comes from. When we describe trends related to a given cloud service provider (CSP), these proportions should help contextualize findings.



Signals **by Cloud Service Provider**

- Amazon Web Services.....57.2%
- Microsoft 365.....30.6%
- Microsoft Azure.....5.7%
- Google Workspace.....4.1%
- Google Cloud Platform.....2.9%

Figure 19: *Signals by Cloud Service Provider*

More than half of cloud-related events came from AWS environments with ~57.21% of all cloud service provider signals. The detections observed may be so high because adversaries can rely on vulnerable or misconfigured EC2 implementations compared to other CSPs.

This year, Elastic added Microsoft 365 (O365) and Google Workspace (GWS) detection signals to our annual data analysis. This provides better visibility of specific services bundled with popular CSPs such as Gmail, Outlook, OneDrive, and more. Overall, Elastic observed that Microsoft 365 trailed behind AWS with ~31% of all signals. However, Azure accounted for only ~6% of detections, potentially suggesting a focus on O365 tenants with common trends such as phishing or valid account compromise.

Elastic also observed that Google Cloud and Google Workspace together accounted for ~6% of all CSP detection signals. While this is relatively low, our threat researchers believe most of the potential campaigns that occur commonly go undetected due to the lack of mature detection rules within these ecosystems. Focusing on tactics and techniques, Elastic Security's prebuilt detection rules are mapped to MITRE's ATT&CK matrix for each CSP.

Tactic	Signal %
Credential Access	47.75%
Defense Evasion	24.44%
Execution	12.30%
Persistence	6.16%
Initial Access	4.98%
Collection	2.08%
Impact	1.09%
Lateral Movement	0.65%
Discovery	0.25%
Privilege Escalation	0.22%
Exfiltration	0.09%

Table 9: *Tactics observed across all Cloud providers*

~45% of all CSP detection signals were related to *Credential Access*. These signals for AWS primarily involved anomalous programmatic retrieval attempts from Secrets Manager, local EC2 host environment variables, and credential file access. Credentials can also be found in poorly reviewed or scrubbed commits to code repositories, like GitHub.

Accounting for 23% of all cloud detection signals is *Defense Evasion*. *System Binary Proxy Execution and Masquerading Execution Paths* are two of the most conventional endpoint techniques, but with CSPs it's commonly associated with disrupting system/environment activity log pipelines to create visibility gaps for SIEM products. For example, disabling AWS CloudTrail logging or changing AWS SQS queuing configurations could disable alerting altogether. In most cases, Elastic observed misconfigurations with AWS security groups, CloudWatch Alarm, and Log stream deletions. For GCP, *Defense Evasion* often involved firewall rule and Pub/Sub topic deletions, which then created visibility gaps in logs.

Elastic observed that adversary playbooks within CSPs involved valid account or vulnerable compute instance compromise, followed by organization/infrastructure discovery, and then existing defense manipulation if possible.

After *Defense Evasion*, *Execution* techniques accounted for ~12% of all cloud detection signals. Another interesting change that Elastic observed this year: the *Exfiltration* technique was rarely observed, dropping from 9% of all signals last year. It is difficult to determine what may have caused this drastic change aside from a deviation in adversary motivation. It's possible that monetary gain via cryptominers deployed to compromised cloud infrastructure was more promising to threat actors than stolen sensitive data to sell on various markets.

Amazon Web Services findings

AWS accounted for nearly 57% of all anomalous cloud-based signal detections Elastic Security Labs observed this year. This year, Elastic observed *Defense Evasion* (~38%), *Credential Access* (~37%) and *Execution* (~21%) as the most common tactics mapped to threat detection signals in AWS.

Compared to 2022, *Credential Access* signals fell nearly 12%, while *Defense Evasion* in AWS increased by ~34%. Elastic attributes this specifically to signals that detect unauthorized security group configuration changes, CloudWatch alarm, and log stream deletion, as well as unauthorized EC2 network access control list (ACL) adjustments.

AWS Signals by Tactic	Signal %
Defense Evasion	38.22%
Credential Access	36.91%
Execution	21.35%
Impact	1.59%
Persistence	0.80%
Lateral Movement	0.73%
Privilege Escalation	0.31%
Collection	0.06%
Initial Access	0.03%

Table 10: *AWS signals by tactic*

> Misconfiguration, lax access control, unsecured credentials, and no PoLP model continue to plague cloud environments where security measures may indeed be under-prioritized while organizations migrate from on-prem to hybrid environments at rapid rates.

While *Credential Access* signals decreased, they still remain prevalent within AWS environments. More specifically, social engineering campaigns, unsecured credentials in logs, or local EC2 host read-only files became high targets for adversaries. While many legitimate users take advantage of both the UI and API access, the same is true of threat actors. We regularly saw threat groups with stolen credentials try to access AWS' Secrets Manager using a variety of programmatic tools, and with accounts that weren't normal for client environments.

AWS Signals by Technique	Signal %
Impair Defenses	37.03%
Unsecured Credentials	36.52%
Cloud Administration Command	21.35%
Data Destruction	1.08%
Indicator Removal	1.07%
Use Alternate Authentication Material	0.77%
Valid Accounts	0.48%
Account Manipulation	0.39%
Steal Application Access Token	0.39%
Service Stop	0.28%
External Remote Services	0.24%
Account Access Removal	0.21%
Modify Cloud Compute Infrastructure	0.07%
Data from Cloud Storage	0.06%
Create Account	0.03%
Data Manipulation	0.02%

Table 11: *AWS Signals by technique*

Regarding techniques, *Cloud Administration Commands* specifically related to AWS' System Manager accounted for ~21% of all signals. Compared to 2022, this is a 42% increase in attempts to execute anomalous commands through the SSM agent which is commonly deployed with default Amazon Machine Images (AMI) in EC2.

Google Cloud Platform findings

About 2.3% of all security events from CSPs came from GCP environments, and ~85% related to *Defense Evasion*. ~11% of all cloud-related threat detections pertained to Data Collection, predominantly due to anomalous behaviors within the Pub/Sub service. GCP's expansive range of services is intertwined with GWS, both sharing infrastructure and IAM functionalities.

GCP Signals by Tactic	Signal %
Defense Evasion	84.69%
Data Collection	10.74%
Impact	2.94%
Persistence	1.50%
Exfiltration	0.13%

Table 12: *GCP signals by tactic*

Undermining defenses emerged as a prevalent technique in our signals, with unauthorized users frequently altering, creating, or deleting GCP firewall rules. Given these rules serve as the fundamental access control layer between Google Compute Engine (GCE) hosts, the implications of tampering are significant.

> Subtle changes to GCP firewall rules or priority reshuffling can inadvertently permit malicious network traffic into environments, leading to cascading security implications.

In GCP environments, enumeration is a preliminary step for adversaries to discern IAM roles, permissions, and infrastructure nuances. Through methods ranging from analyzing local GCP config files in compromised GCE instances to metadata extraction, attackers gain insights into potential privilege levels.

In addition to firewall manipulations, we observed multiple instances where unauthorized user accounts endeavored to alter Virtual Private Cloud (VPC) routing rules applicable to GCE instances. Such modifications can redirect or obstruct network traffic, potentially compromising the security and functionality of the associated cloud infrastructure.

GCP Signals by Technique	Signal %
Impair Defenses	82.91%
Data from Cloud Storage	10.74%
Account Access Removal	1.90%
File and Directory Permissions Modification	1.49%
Data Destruction	1.04%
Account Manipulation	0.73%
Create Account	0.61%
Modify Cloud Compute Infrastructure	0.28%
Valid Accounts	0.16%
Transfer Data to Cloud Account	0.13%

Table 13: *GCP signals by technique*

Google Workspace findings

Earlier in our report, we highlighted new visibility into both Google Workspace and Microsoft 365. These platforms play a pivotal role in evaluating cloud threat detection signals, given their extensive influence over the attack surface of cloud-oriented services. Their offerings—spanning email, storage, communication, and more—are especially tailored for end-user consumption and thus present unique vulnerabilities.

In our analysis of Google Workspace signals, it's crucial to underscore that the majority of anomalous activity signals stem from valid but compromised Workspace accounts, usually with elevated administrative rights or privileges. As such, while Workspace administrators frequently emphasize user security, they must remain acutely aware that their own accounts are prime access points. Here, compromised admin credentials can unlock access to critical configurations like domain-wide application delegation, service accounts, APIs, and other sensitive parameters.

> A breach into a GWS admin account not only presents extensive threat vectors, but also provides potential avenues for lateral movement into GCP environments due to the tight integration between GCP and GWS.

GCP Signals by Tactic	Signal %
Persistence	58.66%
Data Collection	39.55%
Initial Access	1.15%
Defense Evasion	0.58%
Impact	0.06%

Table 14: *Google Workspace signals by tactic*

In our analysis, approximately 98% of all threat detection signals from GWS were attributed to either *Persistence* or *Data Collection* activities. We consistently observed signals where privileged accounts were engaged in relocating users—both new and existing—across different Organizational Units (OU).

GWS employs a hierarchical system for user organization, consisting of OUs and subordinate child OUs. Permissions and roles are typically allocated at the OU level, which means users inherit any associated higher-level privileges. Consequently, an adversary can cunningly gain both *Persistence* and *Privilege Escalation* by introducing a new user account and positioning it within an OU that boasts elevated privileges, such as one associated with a high-tier security group.

GCP Signals by Technique	Signal %
Account Manipulation	58.60%
Data Staged	29.95%
Email Collection	9.61%
Phishing	0.74%
Impair Defenses	0.55%
Valid Accounts	0.41%
Account Access Removal	0.06%
Modify Authentication Process	0.06%
Use Alternate Authentication Material	0.03%

Table 15: *Google Workspace signals by technique*

In GWS environments, Elastic observed a recurring signal pertaining to API access grants especially when domain-wide delegation was activated for third-party applications. This configuration permits GCP service accounts to invoke GWS APIs, effectively acting on users' behalf. Even though such activities can easily masquerade as legitimate operations, it's imperative for GWS administrators to diligently monitor token (OAuth) logs within their domain. By doing so, they can pinpoint and investigate anomalous access patterns, especially stemming from unfamiliar or unexpected applications.

In the context of *Data Collection*, Elastic detected unauthorized users transferring drive ownership to newly-created accounts in multiple occurrences. We also noted the establishment of custom email routes within user Gmail configurations, forwarding to external email addresses—indicative of potential data exfiltration setups..

> *As of this report, Gmail logs containing specifics of inbound and outbound emails are unobtainable via the reports admin API. Consequently, the potential to identify threats like phishing or malware campaigns originating from Gmail remains untapped. Based on our insights, had these logs been accessible, they might have emerged as a predominant threat vector within Google Workspace.*

Microsoft Azure findings

Azure represented about 5.7% of all cloud-related detection signals but remains a popular target for adversaries whose playbook contains utilities specifically for Windows environments. As more on-prem environments become either a hybrid or full CSP deployment, the core of popular Windows services such as Active Directory (AD), SharePoint, and Exchange may migrate as well.

Azure Signals by Tactic	Signal %
Initial Access	60.67%
Persistence	18.87%
Defense Evasion	11.40%
Credential Access	6.89%
Impact	2.09%
Execution	0.06%
Exfiltration	0.03%

Table 16: *Microsoft Azure signals by tactic*

With Microsoft Azure environments, *Initial Access* accounted for ~61% of all threat detection signals and nearly ~19% related to *Persistence*, attributing mainly to multi-factor authentication (MFA) disablement and ownership granting of dedicated Azure applications. This year, *Impact and Credential Access* in Azure fell significantly to ~2% and ~7% respectively.

~53% of *Initial Access* attempts were tied to compromised alid Azure accounts. Notably, a bulk of these signals were generated by a rule looking for anomalous Azure AD sign-ins via PowerShell from on-prem systems, a technique attackers exploit to bypass MFA and session controls. Moreover, we detected unauthorized consent grant attacks on Azure-registered apps, allowing stealthy data access, and Azure Key Vault modifications, potentially exposing sensitive secrets, all flagged by Elastic monitoring.

We observed a high correlation between *Persistence* and *Defense Evasion* tactics in Azure environments. We observed threats disabling MFA and delegating ownership of critical Azure applications, enabling greater undetected access to resources.

Given Azure's app service functions as a host for web applications, RESTful APIs, among others, inherently present internet-facing interfaces. Such exposure amplifies their allure as initial access gateways. Coupled with potential misconfigurations and underlying vulnerabilities, these platforms can inadvertently grant direct virtual machine (VM) access or become conduits for remote code execution via susceptible web applications.

Azure Signals by Technique	Signal %
Valid Accounts	53.16%
Account Manipulation	18.87%
Use Alternate Authentication Material	9.02%
Phishing	8.49%
Unsecured Credentials	6.84%
Resource Hijacking	2.09%
Impair Defenses	1.19%
Exploit Public-Facing Application	0.20%
Command and Scripting Interpreter	0.06%
Steal Application Access Token	0.05%
Transfer Data to Cloud Account	0.03%

Table 17: *Microsoft Azure signals by technique*

Microsoft 365 findings

Credential Access attempts represented about 86% of security events from Microsoft 365 (O365) environments. These were primarily attributed to brute force attempts and password-spraying tactics.

O365 Signals by Tactic	Signal %
Credential Access	86.29%
Persistence	7.16%
Initial Access	4.64%
Lateral Movement	0.77%
Collection	0.56%
Exfiltration	0.29%
Defense Evasion	0.17%
Privilege Escalation	0.13%

Table 18: *O365 signals by tactic*

Persistence signals in O365, contributing to ~7% of total signals, were largely driven by unusual mailbox rights delegations, the assignment of global administrator roles, and significant configuration shifts such as disabling the DKIM signing requirement. These actions can compromise email integrity and allow unauthorized control over key configurations.

A significant chunk of *Initial Access* alerts in O365 related to end-user reports of encountered malware or phishing attempts. This signifies that attackers are using deceptive methods to manipulate users into granting them access.

OneDrive in O365 was often identified as a medium for malware propagation. Once introduced, malicious binaries could be exposed to other O365 users within the same organization.

Drawing parallels with GWS, data collection activities in O365 chiefly emanated from suspicious email forwarding rules established for mailboxes. Such rules redirect sensitive email to external accounts. It's crucial for administrators to maintain strict oversight on mailbox rules and promptly investigate any unanticipated forwarding directives.

O365 Signals by Technique	Signal %
Brute Force	86.29%
Account Manipulation	7.15%
Phishing	4.64%
Taint Shared Content	0.77%
Email Collection	0.56%
Transfer Data to Cloud Account	0.29%
Impair Defenses	0.17%
Domain Policy Modification	0.13%

Table 19: *O365 signals by technique*

Container findings

While not specifically a service provider, Elastic wanted to highlight some threat detection signals pertaining specifically to Kubernetes and the cloud-container ecosystem. *Discovery* accounted for approximately 61% of all Kubernetes-specific signals, which predominantly related to unexpected service account requests that were denied.

K8s Signals by Tactic	Signal %
Discovery	60.93%
Execution	31.32%
Initial Access	7.75%

Table 20: *Kubernetes signals by tactic*

It's worth noting that service accounts in Kubernetes are designed to provide identity to applications running in pods. They're meant to be pre-configured with the appropriate permissions and privileges before any deployment. Any unauthorized requests can indicate misconfigurations or potentially malicious activity.

K8s Signals by Technique	Signal %
Container and Resource Discovery	60.93%
Deploy Container	29.54%
Valid Accounts	7.75%
Container Administration Command	1.78%

Table 21: *Kubernetes signals by technique*

Roughly 31% of all Kubernetes signals were due to the creation of privileged pods within CSP environments in the execution category. Notably, these pods were endowed with privileges equivalent to processes running on the virtual host, giving them unhindered access to all host devices. While certain legitimate operations might need these permissions, it's generally a bad idea to bypass Kubernetes security guidelines purely for operational convenience. Excessive privileges can lead to scenarios where an attacker might exploit container vulnerabilities to gain undue control over the underlying host, jeopardizing the entire node and potential other workloads.

For *Initial Access* within Kubernetes pods, anonymous requests that subsequently gained authorization constituted approximately 8% of all signals. Analysis suggests that these were allowed as anonymous accounts, which, by default, facilitate communication with the API server. To enhance security, it's imperative to either disable or restrict anonymous access and ensure a robust configuration of Role-Based Access Control (RBAC). This ensures that only authenticated and authorized requests are processed by the Kubernetes API server.

Elastic Security Labs observed an unusual uptick in the usage of the **exec** command within pods. This suggests potential attempts to initiate temporary shell sessions or to execute commands inside target pods. While the **exec** command can be a standard tool for Kubernetes administrators, it's crucial to note that there are safer methods for executing commands within a Kubernetes pod. Anomalous **exec** command activity can hint at post-exploitation maneuvers, where attackers aim to expand their influence within the cluster or engage in data exfiltration. As such, monitoring and setting alert thresholds for these anomalies can be pivotal in early threat detection.

Threat profiles

As a part of Elastic Security Labs' dedication to exploring the threat landscape, we have included five major threat profiles developed during the past year. These profiles were chosen based on tracking threats observed in our unique telemetry. The five major activity groups included in this year's Global Threat Report are:

- ICEDID – a globally prevalent malware related to a financially motivated threat
- REF2924 (SIESTAGRAPH, NAPLISTENER, SOMNIRECORD) – espionage activity targeting governments in ASEAN
- REF9134 (JOKERSPY, SWIFTBELT) – a threat phenomenon targeting macOS
- REF2754 (SPECTRALVIPER, P8LOADER, POWERSEAL) – espionage activity targeting domestic financial institutions in Vietnam
- REF9135 (RUSTBUCKET) – a financially motivated activity attributed to DPRK

Threat Naming

At Elastic Security Labs, we use a reference tracking system that clusters activity groups, attack patterns, and intrusion sets together. These clusters are based on specific malware, attack logic, techniques, and in some cases, victimology. They are then given the signifier "REF" and a four-digit number (example: REF1234).

The Diamond Model

For each group listed with a REF identifier, we'll provide a conventional diagram referred to as the Diamond Model. To improve readability, we have pared down overlaps with groups tracked by other vendors, but readers should note that this doesn't indicate agreement or disagreement with those vendors.

We utilize the Diamond Model to describe high-level relationships between the adversaries, capabilities, infrastructure, and victims of intrusions. This model is often used in an intrusion-centric way, but here we employ it with an adversary focus to highlight observations over many incidents.

Terminology

- **Activity group** – Individuals, groups, or organizations believed to be operating with malicious intent.
- **Attack pattern** – A description of the different ways that adversaries attempt to compromise targets.
- **Intrusion set** – Adversarial behaviors and resources with common properties that are believed to be orchestrated by a single organization.

ICEDID

ICEDID is a malware family that was first described in 2017 by IBM X-Force researchers and is associated with the theft of login credentials as well as banking and other personal information. Starting in October 2022, Elastic Security Labs has published several in-depth updates on ICEDID's activity including network infrastructure and analysis techniques, along with a free malware configuration extractor. By providing this research to the community end-to-end, we hope to raise awareness of the ICEDID execution chain, highlight its capabilities, and deliver insights about how it is designed.

Figure 20 depicts the execution chain of the variant that Elastic Security Labs studied and reported on recently. This variant employed multiple stages to load, execute, and establish persistence on a targeted host. You can dive into the full execution chain and a comprehensive analysis of the malware in our research paper, [Thawing the Permafrost of ICEDID].

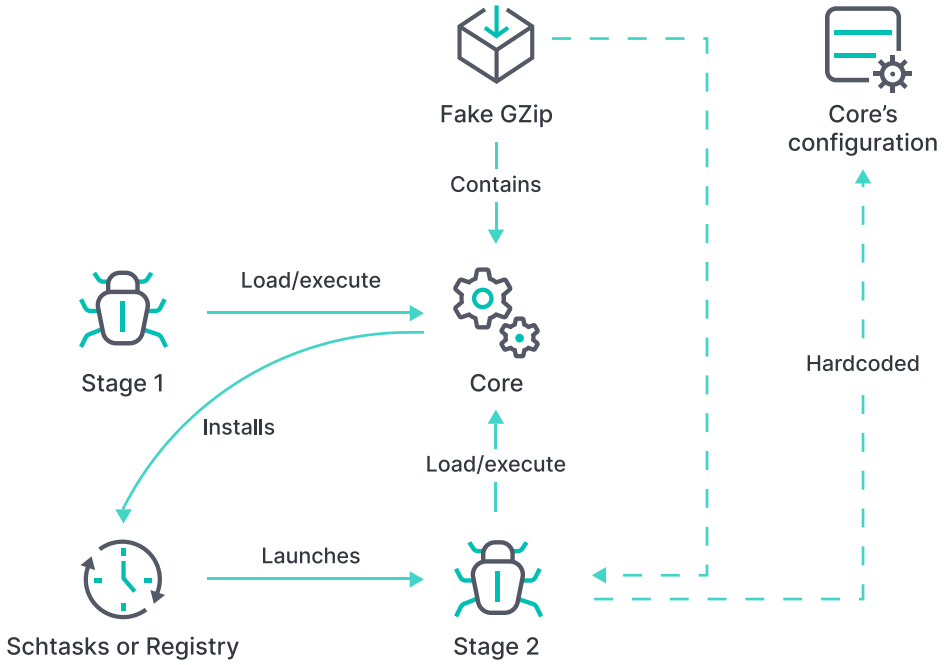


Figure 20: *ICEDID execution chain*

What is the threat?

ICEDID is a full-featured and module banking trojan that targets credentials, personal information, and financial data for theft. ICEDID is known to pack its payloads using custom file formats and a custom encryption scheme.

Elastic Security Labs analyzed multiple ICEDID variants and discovered that they all contained both a loader and a bot payload. A loader can be used to gain an initial foothold onto a targeted host or perform reconnaissance to collect information about the host, like taking note of present security tooling that could impede the intrusion. If the environment isn't overly hostile to the implant, it installs a payload that contains the capabilities used by the threat actor to achieve their campaign objective.

What is the impact?

ICEDID has always been a prevalent family, but it has achieved even more growth since EMOTET's temporary disruption in early 2021. ICEDID has been linked to the distribution of other distinct malware families, including Dark VNC and Cobalt Strike.

What was Elastic's response?

Elastic provides out-of-the-box detections and preventions for ICEDID in the Elastic Security solution. Additionally, Elastic publicly released YARA rules, a detailed campaign and malware analysis, and a configuration extractor. Regular industry reporting, including the research publication released by Elastic, helps mitigate this threat.

Our research has resulted in 15 YARA rules that are focused on identifying various elements of the ICEDID malware. As one example, we released the the following YARA rule to identify the ICEDID bot loader through the reference to its programming database file (**.pdb**), which is observed in the string **E:\source\anubis\int-bot\x64\Release\int-bot.pdb**.

```
rule Windows_Trojan_IcedID {
  meta:
    author = "Elastic Security"
    creation_date = "2023-05-05"
    last_modified = "2023-06-13"
    license = "Elastic License v2"
    description = "IcedID fork
    core bot loader"
    threat_name = "Windows.Trojan.
    IcedID"
    arch_context = "x86"
    os = "windows"

  strings:
    $a = "E:\\source\\anubis\\int-
    bot\\x64\\Release\\int-bot.pdb"
    ascii fullword

  condition:
    all of them
}
```

Learn more

Elastic Security Labs article: I**CEDID's Network Infrastructure is Alive and Well** (October 2022)

Elastic Security Labs article: **ICEDID Configuration Extractor** (October 2022)

Elastic Security Labs article: **Unpacking ICEDID** (May 2023)

Elastic Security Labs Research paper: **Thawing the Permafrost of ICEDID** (March 2023)

Malpedia entry: **ICEDID**

REF2924 – SIESTAGRAPH, SOMNIRECORD, and NAPLISTENER

In December 2022, Elastic Security Labs observed multiple Powershell commands used to collect and export the mailboxes of an Association of Southeast Asian Nations (ASEAN) member from a server for the Foreign Affairs Office. In spite of the diverse security instrumentation observed during this activity, the threat actors were able to achieve complete control of the contested environment.

Expanding our research aperture for this intrusion set, Elastic Security Labs observed REF2924 in three other intrusions from January 2021 through December 2022. Because of the multiple malware samples achieving similar goals, various DLL sideloading observations, and the presence of a likely internet-connected Exchange server; it's possible that there were multiple threat actors or threat groups working independently or in tandem with each other.

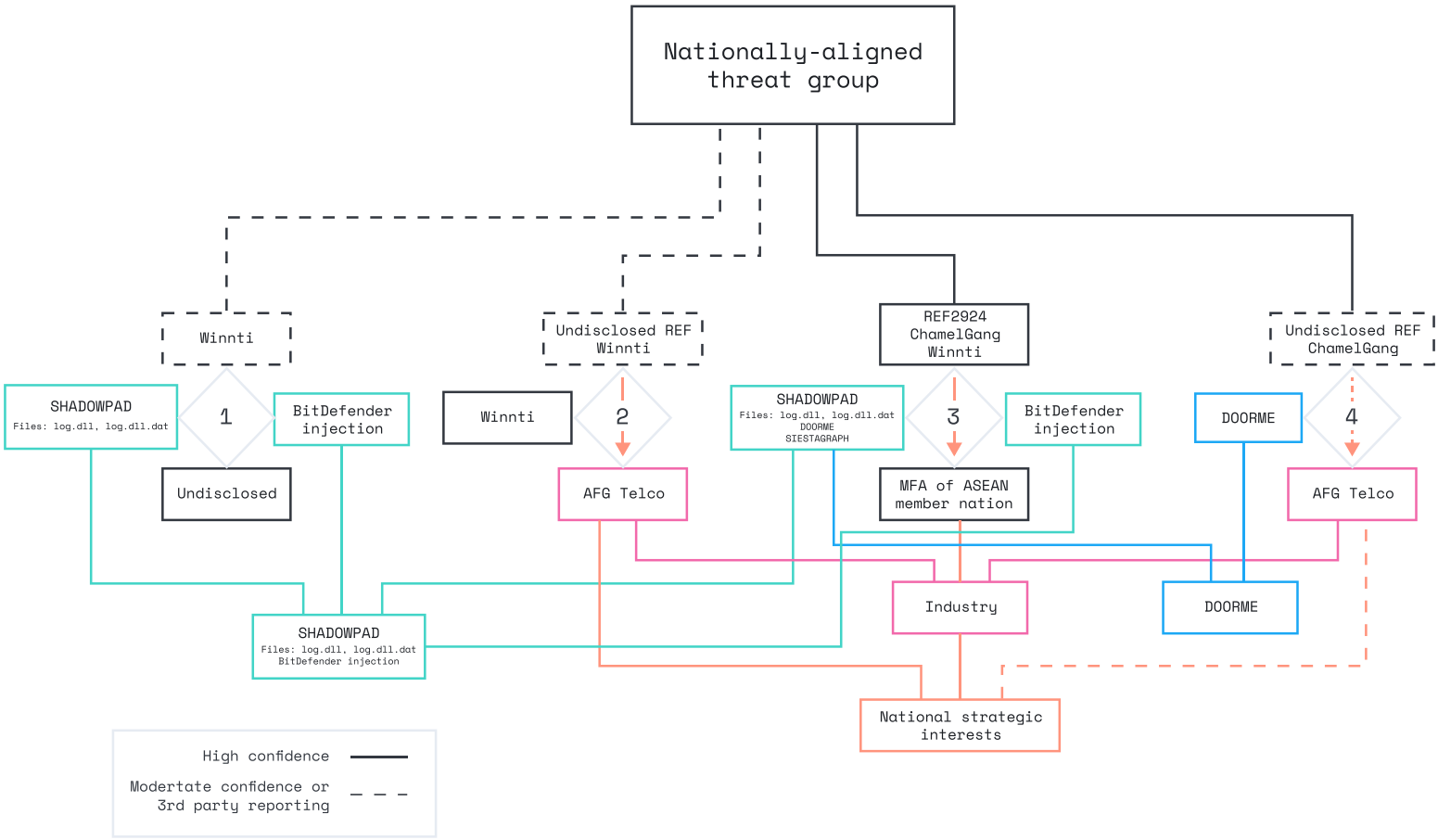


Figure 21: *REF2924 intrusion intersections and associations*

What is the threat?

REF2924 is a diverse intrusion set using:

- The backdoored Internet information services (IIS) module called DOORME
- The backdoor called SHADOWPAD
- A previously undiscovered HTTP listener which we named NAPLISTENER
- A previously undiscovered backdoor which we named SOMNIRECORD
- A previously undiscovered implant that uses Microsoft's GraphAPI for C2 which we named SIESTAGRAPH.

DOORME, SIESTAGRAPH, and SHADOWPAD each implement different functions that can be used to gain and maintain unauthorized access to an environment. It is important to note that these backdoors can be used to steal sensitive information, disrupt operations, and gain a persistent presence in a victim environment.

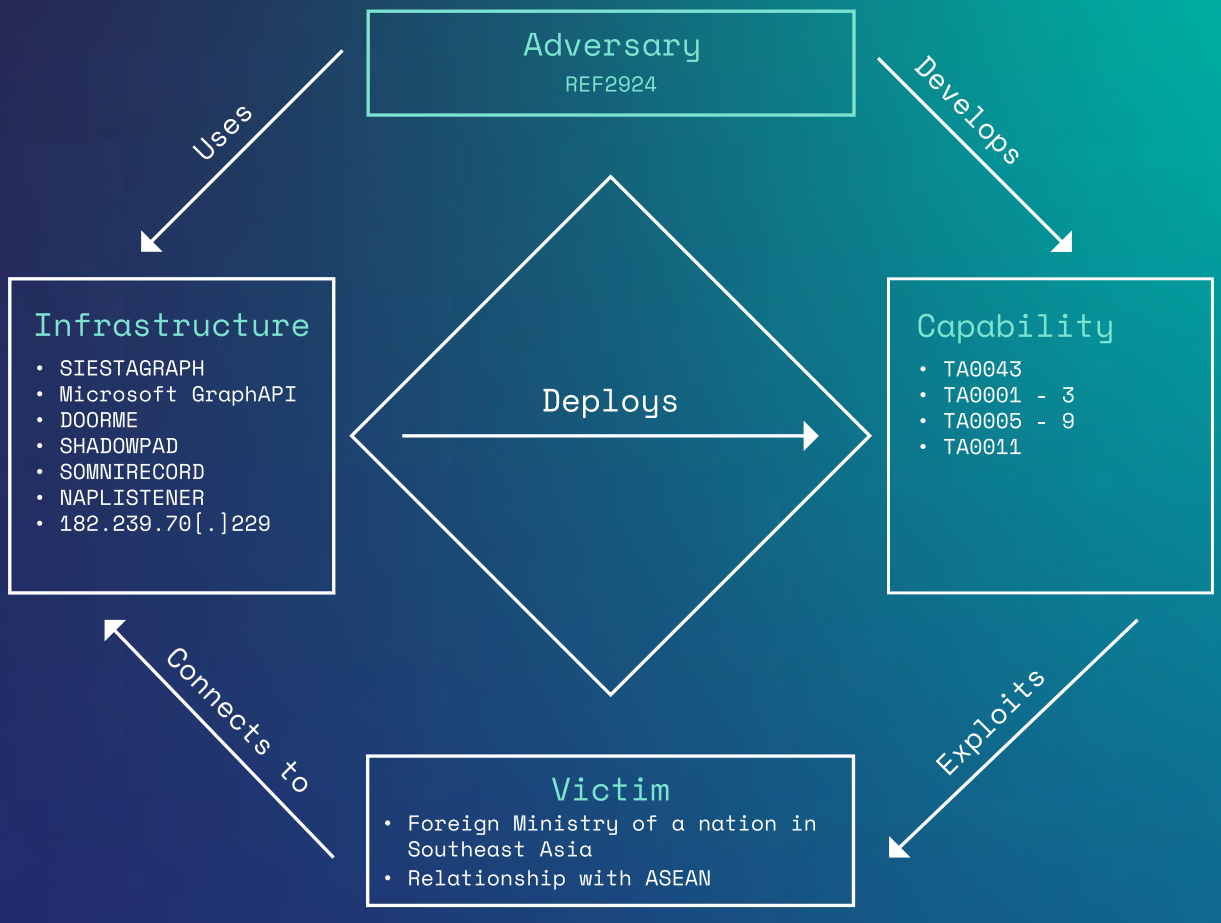


Figure 22: *REF2924 Diamond Model*

What is the impact?

The REF2924 intrusion set was used to successfully take control of Exchange servers, Domain Controllers, and workstations, exfiltrated targeted user mailboxes, deployed web shells, performed internal reconnaissance, and collected credentials. We assess with high confidence that this threat operates at the direction of the People's Republic of China.

What was Elastic's response?

The Elastic Security Labs team detailed all of the observed malware used by the intrusion set, extracted observables for endpoint and network filtering, and produced new malware signatures for identification and mitigation of the intrusion set.

Our research into REF2924 has resulted in six YARA rules that are focused on identifying the different malware observed in this intrusion set. As one example, we released the YARA rule below to identify the SIESTAGRAPH backdoor through the use of strings present in the malware.

```
rule Windows_Trojan_SiestaGraph {
meta:
author = "Elastic Security"
creation_date = "2022-12-14"
last_modified = "2022-12-15"
license = "Elastic License v2"
os = "Windows"
arch = "x86"
category_type = "Trojan"
family = "SiestaGraph"
threat_name = "Windows.Trojan.SiestaGraph"

strings:
$a1 = "downloadAsync" ascii nocase fullword
$a2 = "UploadxAsync" ascii nocase fullword
$a3 = "GetAllDriveRootChildren" ascii fullword
$a4 = "GetDriveRoot" ascii fullword
$a5 = "sendsession" wide fullword
$b1 = "ListDrives" wide fullword
$b2 = "Del OK" wide fullword
$b3 = "createEmailDraft" ascii fullword
$b4 = "delMail" ascii fullword

condition:
all of ($a*) and 2 of ($b*)
}
```

Learn more

Elastic Security Labs article: **SIESTAGRAPH: New implant uncovered in ASEAN member foreign ministry** (December 2022)

Elastic Security Labs article: **NAPLISTENER: more bad dreams from developers of SIESTAGRAPH** (March 2023)

Elastic Security Labs article: **Not sleeping anymore: SOMNIRECORD's wake-up call** (March 2023)

Malpedia entry: **SIESTAGRAPH**

Malpedia entry: **NAPLISTENER**

REF9134 - JOKERSPY and SWIFTBELT

In May 2023, Elastic Security Labs observed a new intrusion set with prior access to a prominent Japanese cryptocurrency exchange service provider. This intrusion set, which we named REF9134, included a previously undiscovered macOS binary later dubbed JOKERSPY, a previously undiscovered Python backdoor named sh.py, an open source reconnaissance tool called SWIFTBELT, and abused native macOS security features.

What is the threat?

REF9134 is an intrusion set that abuses the macOS Transparency, Consent, and Control (TCC) database while leveraging three main malware families: JOKERSPY, sh.py, and SWIFTBELT. As observed by Elastic Security Labs, the intrusion set works in a specific order:

1. JOKERSPY, a self-signed multi-architecture binary written in Swift, is used to evaluate system permissions in preparation for later-stage malware

2. sh.py, a Python backdoor, is used to deploy and execute post-exploitation commands and tools

3. SWIFTBELT, a macOS malware, performs multiple enumeration and data collection functions using Swift language.

```
v3 = type metadata accessor for XProtectCheck();

xCheck = swift_allocObject(v3, 16LL, 7LL);

specialized XProtectCheck.SystemIdleTime() ();

specialized XProtectCheck.getTopWindowApp() ();

specialized XProtectCheck.isScreenLocked() ();

specialized XProtectCheck.checkFullDiskAccessPermission() ();

v4 = GPreflightScreenCaptureAccess(v3);
```

Figure 23: *JOKERSPY evaluating system permissions*

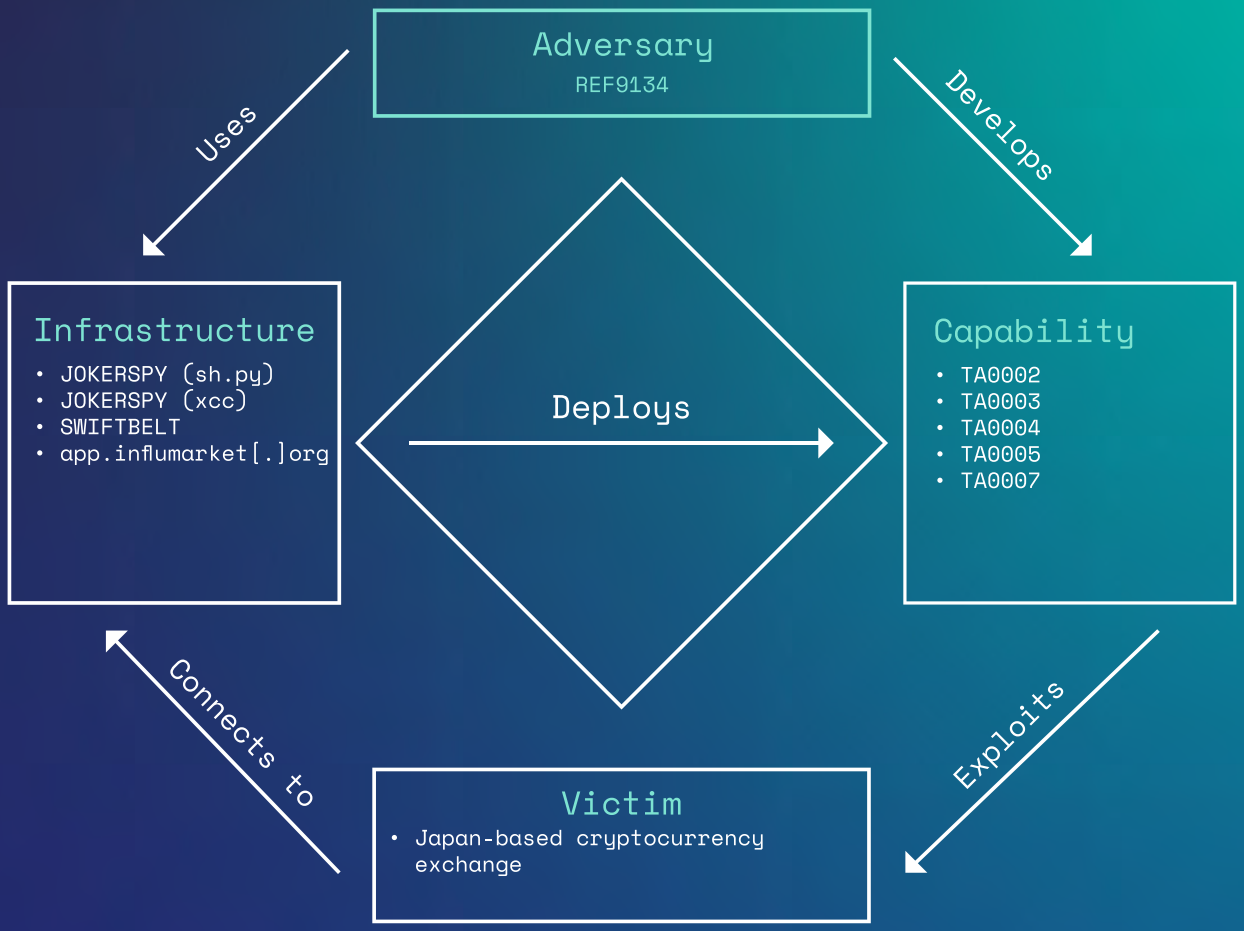


Figure 24: *REF9134 Diamond Model*

What is the impact?

Leveraging the REF9134 intrusion set, the threat actor was able to gain and maintain persistence in the contested network — a cryptocurrency in Japan. They do not appear to have successfully advanced their campaign past the deployment of SWIFTBELT before being evicted by the system owners.

What was Elastic's response?

Elastic released an analysis product highlighting the execution and persistence mechanisms leveraged by the REF9134 intrusion set. Additionally, we released YARA rules for malware we observed in the intrusion set – namely endpoint prevention rules and host and network atomic indicators.

Our research into REF9134 has resulted in two YARA rules that are focused on identifying the different malware observed in this intrusion set. As one example, we released the follow YARA rule to identify the JOKERSPY tool through the use of strings present in the malware.

```
rule Macos_Hacktool_JokerSpy {
  meta:
        author = "Elastic Security"
        creation_date = "2023-06-19"
        last_modified = "2023-06-19"
        license = "Elastic License v2"
        os = "MacOS"
        arch = "x86"
        category_type = "Hacktool"
        family = "JokerSpy"
        threat_name = "Macos.Hacktool.JokerSpy"

strings:
  $str1 = "ScreenRecording: NO" fullword
  $str2 = "Accessibility: NO" fullword
  $str3 = "Accessibility: YES" fullword
  $str4 = "eck13XProtectCheck"
  $str5 = "Accessibility: NO" fullword
  $str6 = "kMDItemDisplayName = *TCC.db" fullword

condition:
  5 of them
}
```

Learn more

Elastic Security Labs article: **Initial Research exposing JOKERSPY** (June 2023)

Malpedia entry: **JOKERSPY**

REF2754 - SPECTRALVIPER, P8LOADER, AND POWERSEAL

In June 2023, Elastic Security Labs uncovered an intrusion set targeting large Vietnamese companies. This intrusion set is tracked as REF2754 and included three new malware samples: SPECTRALVIPER, P8LOADER, and POWERSEAL.

Additional analysis of REF2754 connected it to another intrusion set tracked as REF4322, which included the malware PIPEDANCE and PHOREAL. REF4322 was highlighted in Elastic's 2022 Global Threat Report.

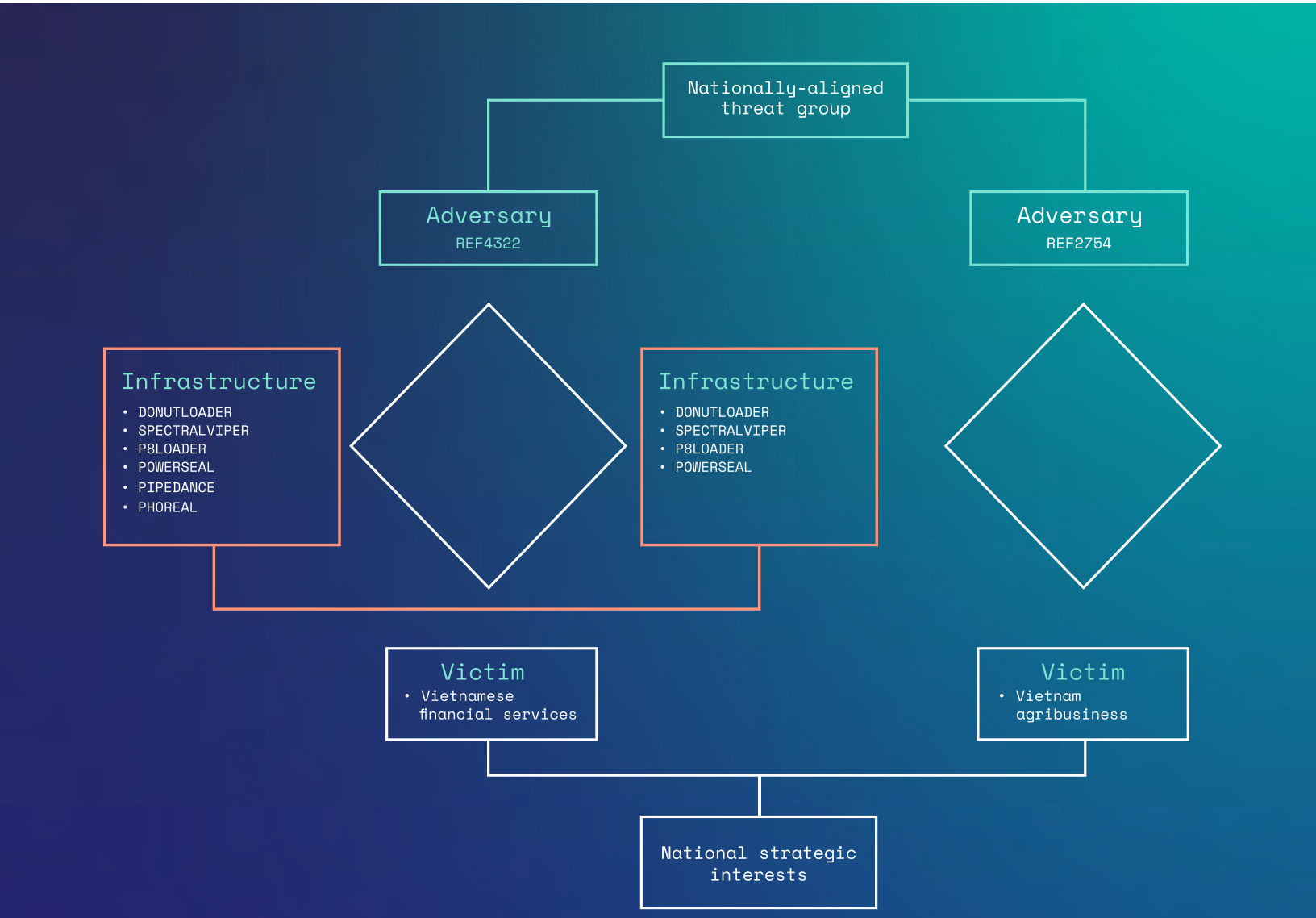


Figure 25: *REF2754 intrusion intersections an associations*

What is the threat?

REF2754 is an intrusion set that includes LOLBAS techniques, a known open source memory injector, and three new malware families that were first identified by Elastic Security Labs. The new malware makes up the final part of the intrusion set execution flow and consists of a backdoor that Elastic named SPECTRALVIPER, a x64 PE loader that we named P8LOADER, and a PowerShell runner we named POWERSEAL.

What is the impact?

The REF2754 intrusion set was observed in a campaign against large nationally important public companies within Vietnam. The threat actors were able to maintain residence within the contested network for several months. The intrusion set has been attributed to a Vietnamese-based threat group, aligning with the Canvas Cyclone/APT32/OceanLotus threat groups.

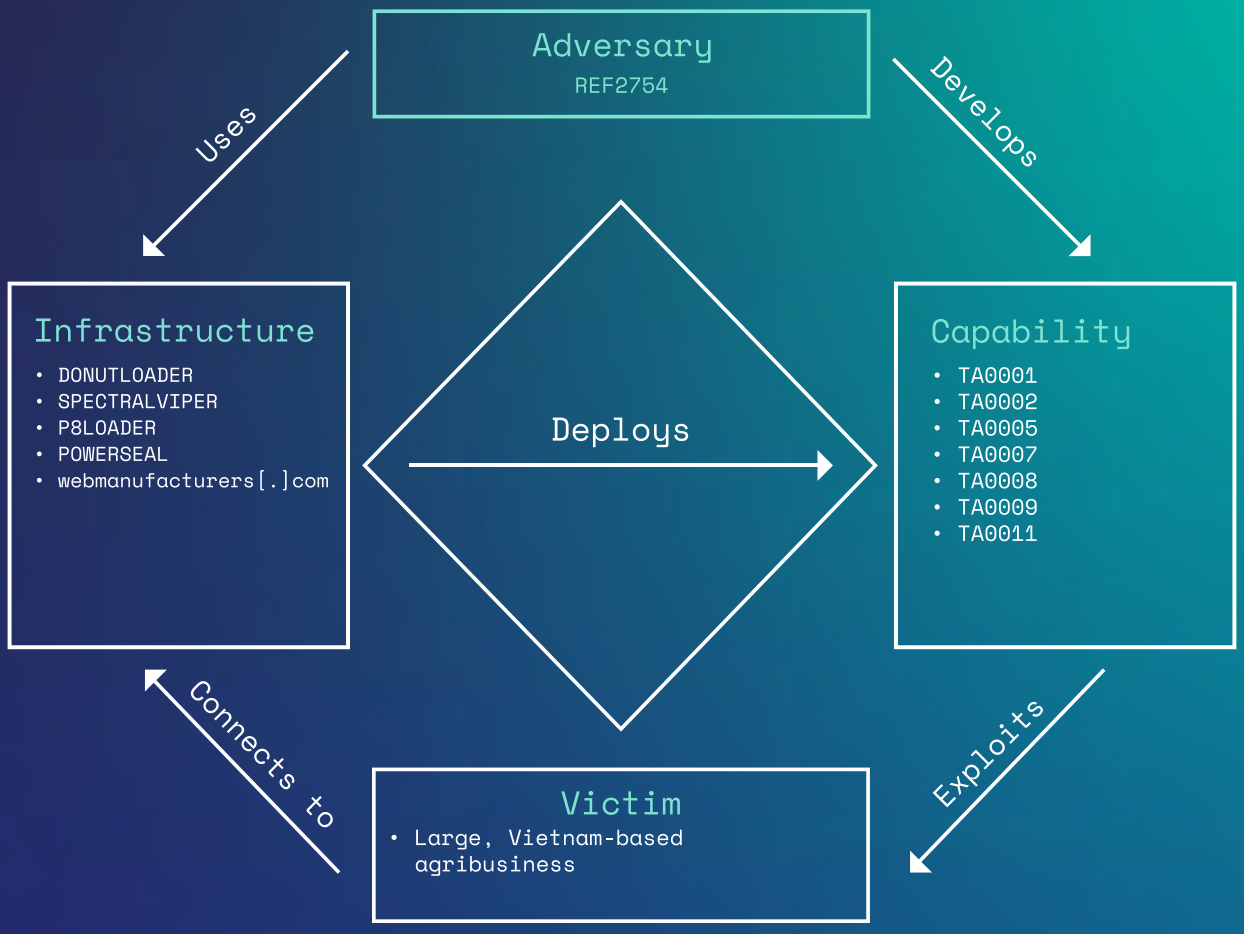


Figure 26: *REF2754 Diamond Model*

What was Elastic's response?

Elastic has publicly released detailed malware and campaign analysis, YARA signatures, and endpoint protections to detect and prevent all malware in this intrusion set. Additionally, we released all atomic indicators observed in this intrusion.

Our research into REF2754 has resulted in five YARA rules that are focused on identifying the different malware observed in this intrusion set. As one example, we released the following YARA rule to identify the POWERSEAL PowerShell runner through the use of strings present in the malware.

```
rule Windows_Trojan_PowerSeal {
  meta:
        author = "Elastic Security"
        creation_date = "2023-03-16"
        last_modified = "2023-05-26"
        license = "Elastic License v2"
        os = "Windows"
        arch = "x86"
        category_type = "Trojan"
        family = "PowerSeal"
        threat_name = "Windows.Trojan.PowerSeal"

  strings:
        $a1 = "PowerSeal.dll" wide fullword
        $a2 = "InvokePs" ascii fullword
        $a3 = "amsiInitFailed" wide fullword
        $a4 = "is64BitOperatingSystem" ascii fullword

  condition:
        all of them
}
```

Learn more

Elastic Security Labs article: **Elastic charms SPECTRALVIPER** (June 2023)

Malpedia entry: **SPECTRALVIPER**

REF9135 - RUSTBUCKET Variant

In June 2023, the Elastic Security Labs team detected a new variant of the RUSTBUCKET malware, a family that has been previously attributed to the BlueNorOff group by Jamf Threat Labs in April 2023. The research includes REF9135's use of RUSTBUCKET for sustained operations at a cryptocurrency payment services provider, but Elastic Security Labs specifically identified a variant of RUSTBUCKET that was previously undetected.

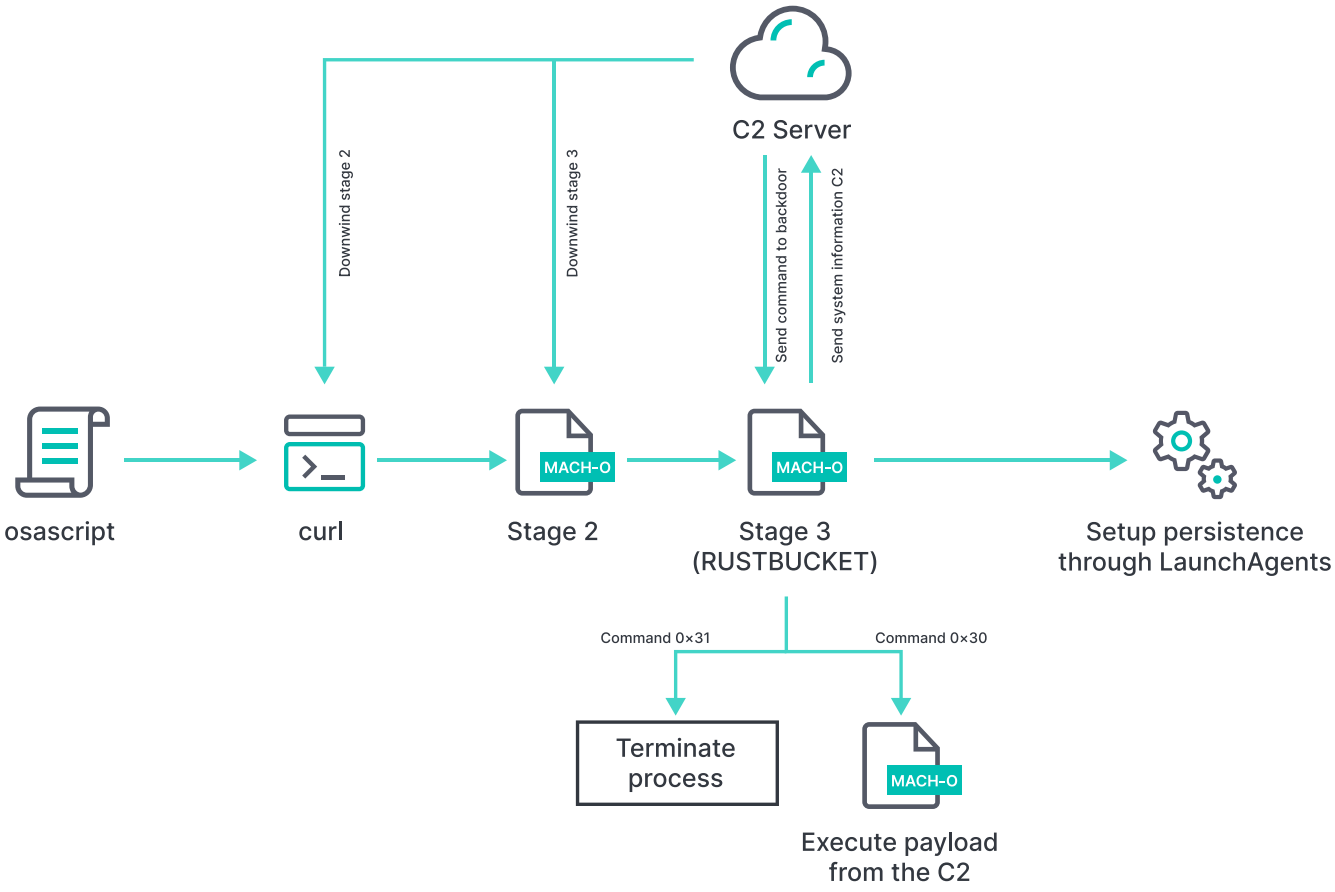


Figure 27: *REF9135 execution flow*

What is the threat?

This variant of RUSTBUCKET, a malware family that targets macOS systems, adds persistence capabilities not previously observed and, at the time of reporting, was undetected by VirusTotal signature engines. This also confirms that the RUSTBUCKET malware family is in active development.

What is the impact?

The research into REF9135 used host, binary, and network analysis to identify and attribute intrusions with high confidence to the Lazarus Group, as observed by this research team and other intelligence groups. The Lazarus Group is a cybercrime and espionage organization operated by the Democratic People's Republic of North Korea (DPRK).

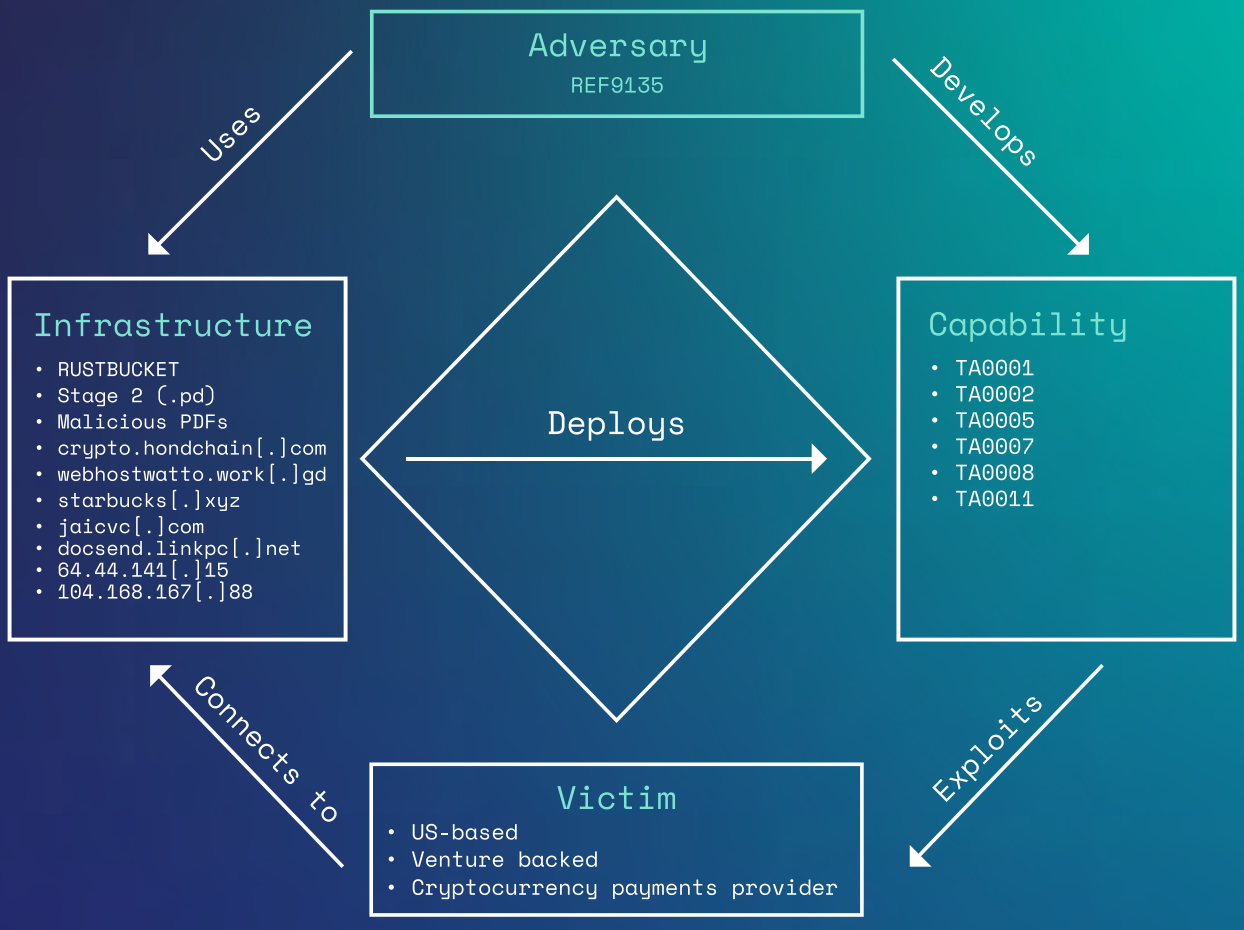


Figure 28: *REF9135 Diamond Model*

What was Elastic's response?

To protect endpoints and cloud environments, the Elastic Agent behavioral and prebuilt detection rules provide protection and visibility for users. We also released an additional behavioral rule for persistence techniques observed in this intrusion, hunting queries, host and network indicators we observed, along with one YARA rule to identify the RUSTBUCKET variant. Outlined below, this rule identifies the new variant through the use of strings present in the malware.

```
rule MacOS_Trojan_RustBucket {
  meta:
        author = "Elastic Security"
        creation_date = "2023-06-26"
        last_modified = "2023-06-29"
        license = "Elastic License v2"
        threat_name = "MacOS.Trojan.RustBucket"
        arch = "x86"
        os = "macos"

  strings:
        $user_agent = "User-AgentMozilla/4.0 (compatible; MSIE 8.0;
        Windows NT 5.1; Trident/4.0)"
        $install_log = "/var/log/install.log"
        $timestamp = "%Y-%m-%d %H:%M:%S"

  condition:
        all of them
}
```

Learn more

Elastic Security Labs article: [The DPRK strikes using a new variant of RUSTBUCKET](#) (July 2023)

Reflecting on last year's forecasts

In our inaugural threat report, Elastic Security Labs offered six forecasts for 2023 based on trends, correlations, and our visibility into the dynamic global threat landscape. In remaining aligned with our value of transparency, we wanted to spend some time reflecting on the accuracy of our estimations.

1. **Adversaries will continue to abuse built-in binary proxies to evade security instrumentation**
   Verdict: We were right, but this was a safe suggestion

   At the time of analysis, we observed that threat actors relied on built-in binary proxies frequently during intrusions, but this may have been too safe a forecast. Whether it was using **rundll32.exe** to load a malicious DLL containing a reverse shell or **mshta.exe** to run a VBscript containing reconnaissance commands, binary proxies aren't going anywhere soon. A recent example of this was **LOBSHOT**, which used a binary proxy to load a backdoor into memory.

2. **LNK and ISO payloads will replace more conventional script and document payloads**
   Verdict: We were incorrect

   When Microsoft neutralized the use of macro-enabled Word documents, we felt LNK and ISO attachments would become the de jour methodology. While this approach did experience a brief period of popularity, more conventional approaches like malicious HTML attachments containing links to adversary infrastructure were much more commonly adopted. This type of phishing attachment has been used to deploy the **YIPPHB** dropper, one component of the NJRAT infection chain.

3. **Valid IAM accounts will continue to be a target for adversaries**
   Verdict: We were right, but this was a safe suggestion

   As the shortest possible path to data, a valid account is the ideal target for threats whether their objectives are financial, disruptive, or espionage oriented. Throughout 2023, threat groups have sought valid credentials with a **strong cloud bias**.

4. **If service accounts managed by major CSPs are not configured with least-privilege permissions, they will be aggressively targeted by adversaries**
   Verdict: We were right, but this was a safe suggestion

   While we can concede that this forecast may have been too safe, it remains an important risk for many organizations who may configure accounts with unnecessarily broad access to cloud resources and enterprise data. Threat groups have benefited from *Privilege Escalation* flaws in common functions in order to **further abuse** overly privileged service accounts.

5. **Cloud deployed Linux virtual machines for backend DevOps may see an increase in account enumeration and Credential Access attacks**
   Verdict: Less than we anticipated

   While we do occasionally see evidence of it, it seems that threat actors lean on more reliable means of account compromise. Phishing, as one example, was observed more often than prior years. Malware designed to steal credentials goes so far as to target GitHub CodeSpaces, an enterprise cloud-based development platform where API keys, tokens, and other credentialed materials may be present.

6. **If organizations over invest in detection capabilities but do not also support mitigation, they will struggle with responding to all categories of threats**
   Verdict: We were correct

   Unfortunately, we saw this numerous times throughout the year, as enterprises resisted adversaries with various degrees of effectiveness. Security teams need agency and capability to counter a high-tempo threat, and it can be very challenging to keep up otherwise.

Forecasts & recommendations

Elastic Security Labs considered trends, correlations, and currently unpublished research materials to assemble these forecasts for the coming year. Two phenomena forecast in last year's report, related to the prevalence of *Defense Evasion* investments and *Cloud Credential Access*, reappear. In each of these cases we've tried to be more specific, a challenge we'll measure ourselves against next year. With each forecast we've included a related call to action for security teams to implement.

Forecast 1:

Defense Evasion is going to remain the top investment, and tampering will supersede masquerading

The dynamics of the security industry have produced impressive endpoint prevention features, and adversaries know that evading those features is the surest way to achieve their objectives. We believe this will result in masquerading finally losing ground to tampering, and we've observed this in the wild with capabilities like Bring Your Own Vulnerable Driver (BYOVD).

Call-to-action

Enterprises should evaluate the tamper-resistant nature of their endpoint security sensors, and consider monitoring projects like living off the land Drivers which tracks the many vulnerable device drivers used to disable security technologies.

Forecast 2:

The malware-as-a-service (MaaS) model will become more popular

This year we saw several financially motivated threat communities adopt or offer MaaS capabilities. MaaS shops are going to sell to a huge population of affiliates, resulting in large numbers of malware variants. With heavy investments across the board in *Defense Evasion*, we can expect *Obfuscation* and *Masquerading* to play important roles. Given the use of benign applications to smuggle malware into organizations like BLISTER, we may see broader adoption of that methodology.

Call-to-action

The rapid development lifecycle of MaaS requires enterprises to develop similarly rapid security functions with broad visibility of low-level behaviors. An example of this is Meterpreter, a payload for the offensive security tool Metasploit that provides an interactive shell to the attacker, which may be inconsistently identified by antivirus products due to the breadth of features it contains. Endpoint sensors that capture interprocess behaviors, filesystem interactions, and network data are essential to expose these previously undiscovered threats.

Forecast 3:

Adversaries will become more reliant on open-source communities for implants, tools, and infrastructure

Throughout the year, we noted that threats of all kinds were using code from open sources. Whether a legitimate library like OneDriveAPI, a tool like SharpShares, or an implant like Sliver, it is a near certainty that adversaries will continue to take advantage of publicly exposed projects.

Call-to-action

Organizations should scrutinize direct downloads from code-sharing websites and consider limiting access. While this wouldn't impact code reused by threat actors, it could prevent precompiled binaries or portable scripts from facilitating a compromise. Enterprises should evaluate their visibility of emerging adversarial frameworks.

Forecast 4:

Cloud credential exposure will be a primary source of data exposure incidents

During the prior two years, adversaries have demonstrated a clear focus on stealing credentials for enterprise cloud solutions. With stolen credentials, it is a simple process to authenticate and access cloud storage, which is not often segmented in large organizations, to stage malware or store pilfered data. In addition to data theft, we believe it is likely that exposed cloud computing credentials will increase the prevalence of coinminers and other malware.

Call-to-action

Least privilege accounts and robust authentication mechanisms can be augmented by monitoring user-entity behaviors, solutions that may depend on reasonable segmentation of data.

Forecast 5:

Excessively privileged Kubernetes pods will compound the damage of container vulnerabilities

More than 30% of the Kubernetes signals we saw in last year's Global Threat Report were the result of creating privileged pods with full host device access, a significant risk to the node and any workloads. Adversaries may leverage vulnerabilities to further exploit this configuration, deploying additional malware or stealing sensitive data.

Call-to-action

Security teams should work with operations teams to configure and deploy containers with the least amount of necessary privileges, and engage in runtime monitoring to determine if the privilege characteristics of pods are changing abruptly, which may indicate a compromise.

Conclusion

For the second year in a row, adversaries have shown us just how important *Defense Evasion* and *Credential Access* are to achieving their objectives. Threat actors continue to demonstrate their priority of getting data without getting caught, and we've observed that underlying excessive privileges, often defaults, make this much easier for the adversary. Security professionals need to invest in technologies that make these tactics harder.

We can expect to see adversaries attacking security technologies, to the extent that operating systems permit such behaviors, leveraging vulnerable device drivers and other methods to disable or circumvent controls. Meanwhile, adversaries with the lowest risk tolerance are moving to the edge or directly dealing with cloud-resident data to minimize costs associated with being discovered.

As far as we've come, we need to continue pursuing our goal of protecting the world's data from attack. In discovering several related espionage-motivated threat groups, we learned how much they depend on open sources for their capabilities; organizations need to understand that the barriers to entry are gone, and public projects published pseudo-anonymously are empowering threats of all kinds. This also speaks to the essential role that threat research plays in discovering, exposing, contextualizing, and mitigating these phenomena.

Elastic discovered this firsthand, as the last year saw us pushing mitigations for more than a dozen never-before-seen malware families, shipping numerous signatures for general-purpose capabilities, and publishing nearly 50 research articles. But none of this would be possible without the telemetry from Elastic Agent, which gave us visibility into low-level threat activity that may not yet be detected or stopped.

The only way to change the threat landscape is to deliberately attempt to change it. The team here at Elastic Security Labs does that by democratizing access to knowledge and resources, developing and releasing no-cost technologies that empower defenders, and actively opposing threat actors with our powerful global instrumentation.

Step by step, it's getting better. That's what we're here to help you achieve.

Learn about **Elastic Security** and protect against the threats covered in this report (and other vulnerabilities) by visiting our **Elastic Security Labs** page. You can also **follow us on X** to see when we release news-breaking threat research.

Find more research:

www.elastic.co/security-labs

Follow us on X (formerly Twitter):

@ElasticSecLabs