
CyberProof®
A UST Company
MITRE playbook
Understanding today's threat landscape using MITRE ATT&CK frameworks

Table of Contents

Report Guide

The report is divided into chapters, with each chapter representing a MITRE tactic.

Under each tactic, statistics of the techniques linked to it are presented, as well as statistics of their sub-techniques, if relevant.

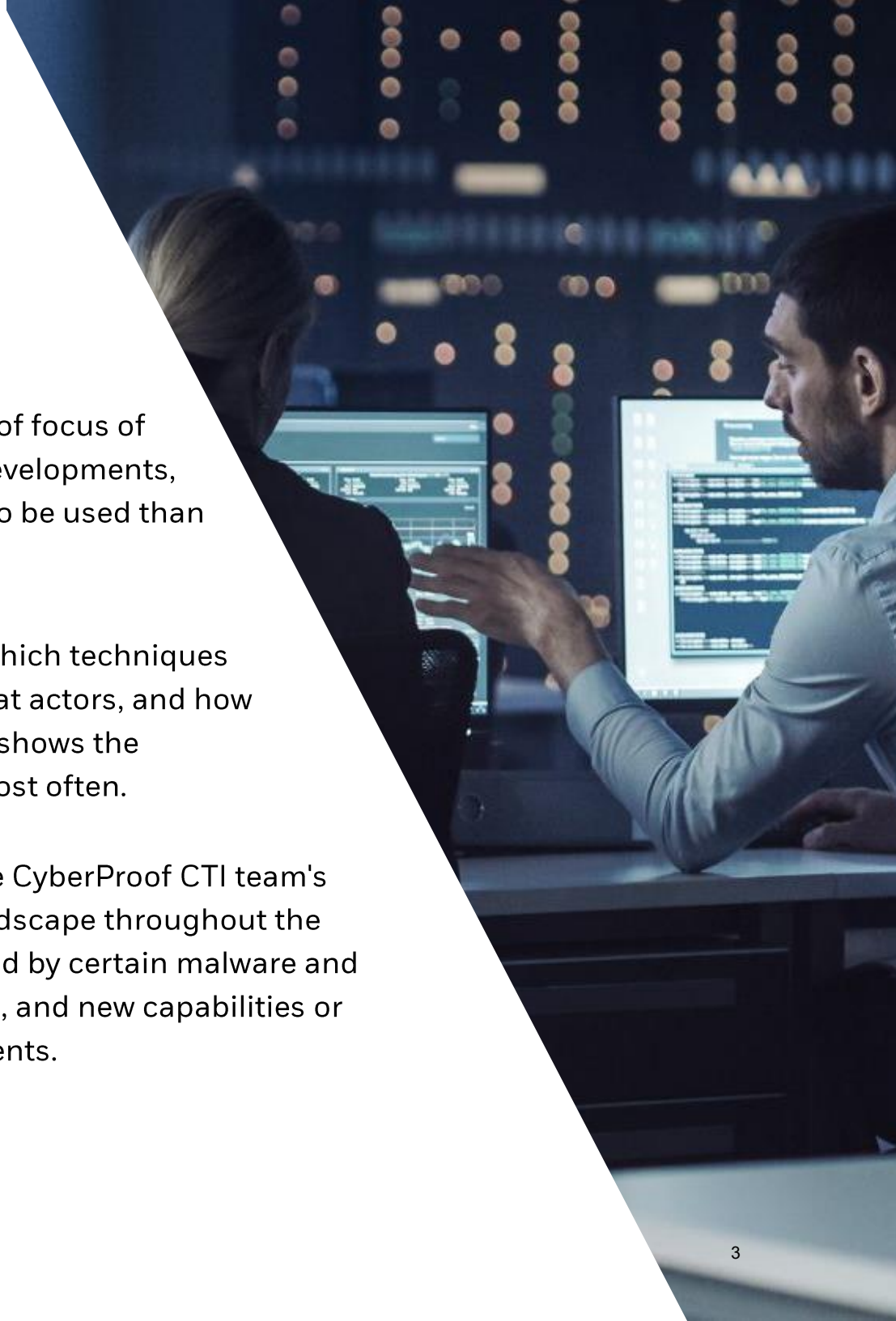
After each statistics graph comes an explanation of the techniques presented, the story behind the numbers, and their connection to events that happened in the last year.

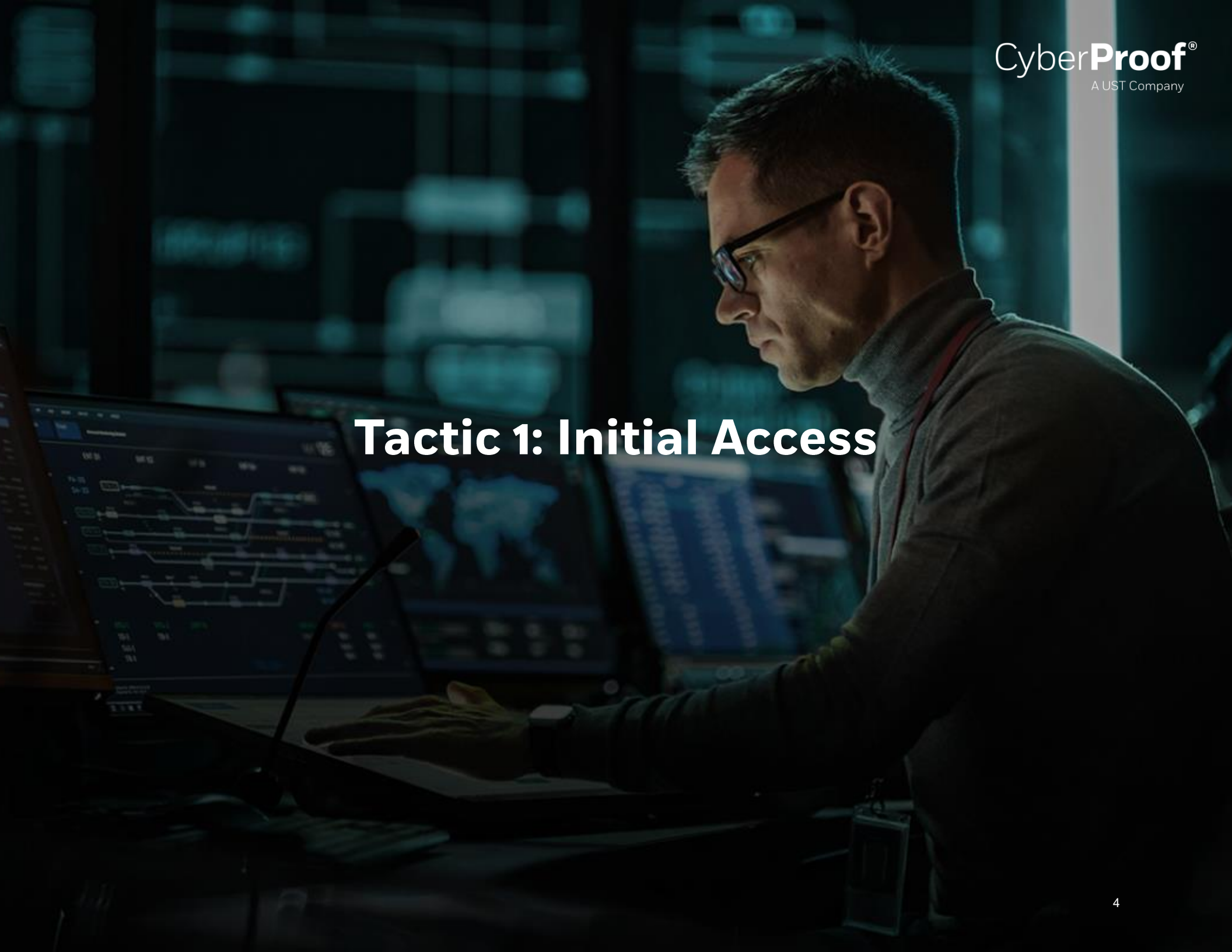
Introduction

This report looks to tell the story of the main areas of focus of threat actors, how external factors affected their developments, and what makes one attack technique more likely to be used than another.

This story is told by analyzing each MITRE tactic, which techniques and sub-techniques were the most favored by threat actors, and how they are seen in action. Simultaneously, the report shows the challenges that cyber defenders had to mitigate most often.

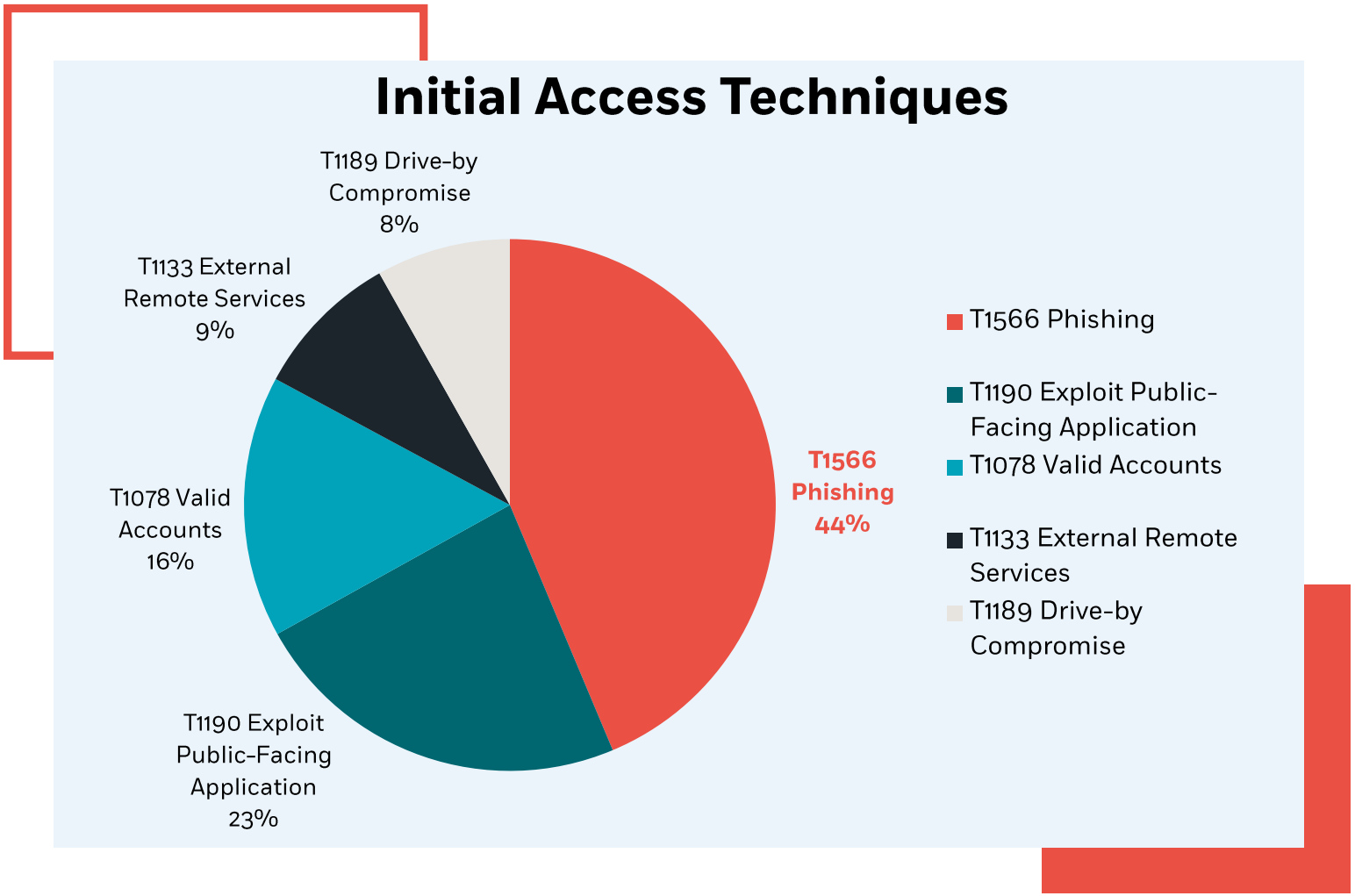
The data analyzed in this report is comprised of the CyberProof CTI team's analysis of relevant developments in the threat landscape throughout the span of one year – such as new campaigns executed by certain malware and ransomware types, recently-discovered techniques, and new capabilities or attack vectors that could potentially impact our clients.





Tactic 1: Initial Access

Tactic 1: Initial Access – Technique Statistics



The leading technique used as initial access in campaigns reported by the CyberProof CTI team was **T1566 Phishing**, making up almost half of the techniques used throughout the year.

T1566 Phishing

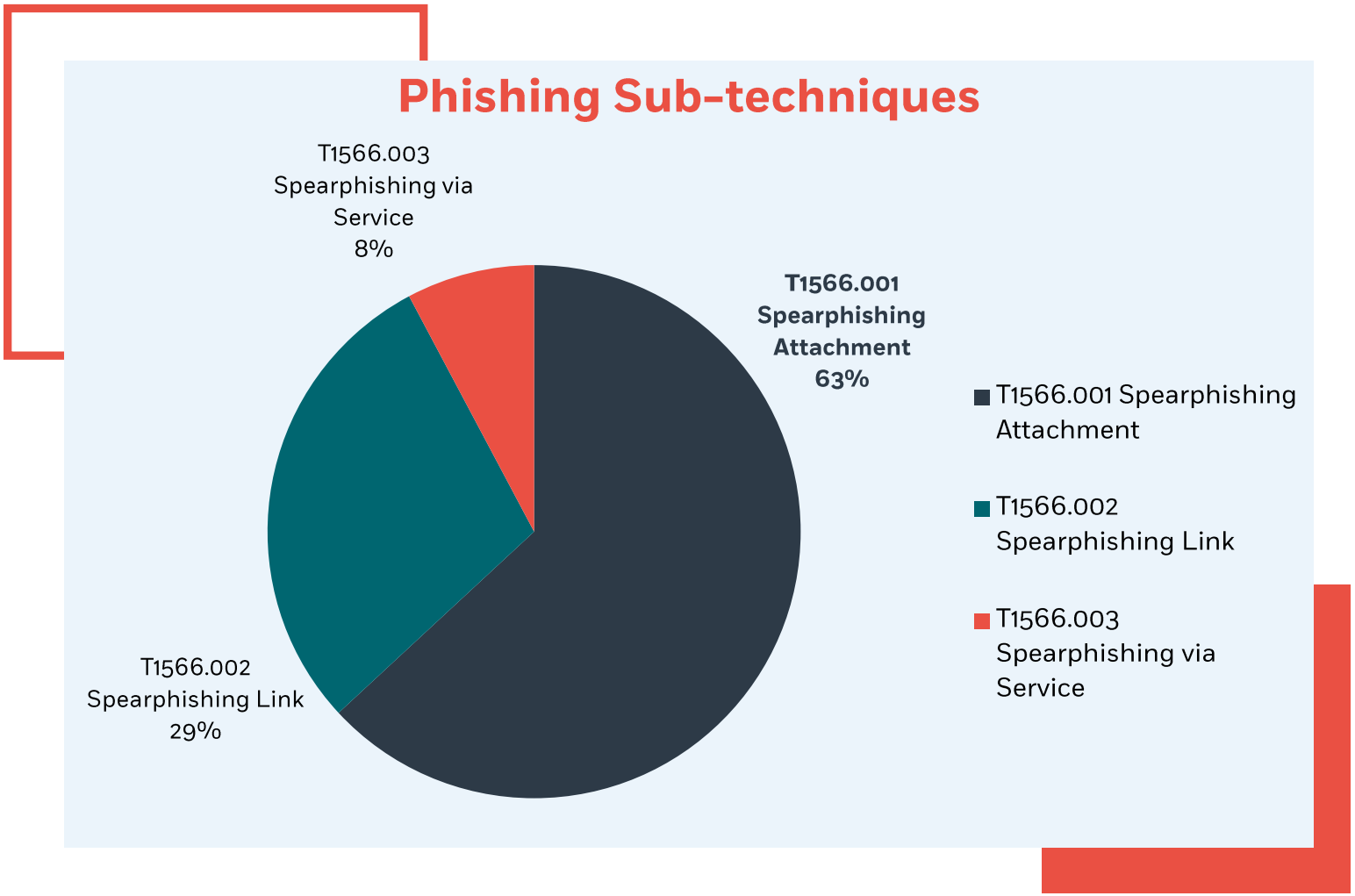
Phishing has become one of the most prevalent methods used by threat actors to launch attacks against unsuspecting individuals and enterprises, including delivering malware and ransomware payloads and luring victims to reveal sensitive information such as login credentials or financial details.

Phishing attacks typically involve the use of deceptive emails, text messages, or social media messages, which in most cases, are carefully crafted to trick the recipient into clicking on a malicious link or downloading an infected file.

Once the link or file is opened, the malware is then installed on the victim's device, giving the attacker control over the system and access to sensitive data, and in cases of ransomware attacks, the ability to encrypt the victim's data and demand payment in exchange for the decryption key.

Phishing emails are often used to deliver the initial payload of ransomware, with attackers using social engineering tactics to lure the victim into opening a malicious attachment or clicking on a link that leads to the installation of the ransomware.

These attacks can be particularly devastating, as they can result in the destruction of important data, financial loss, and damage to the victim's reputation.



By taking a deeper dive into Phishing Sub-techniques, we identified that **Spearphishing Attachment** is the sub-technique most utilized by attackers.

T1566.001 Spearphishing Attachment Sub-technique

Changes that have occurred in the technological field this year have influenced the fact that Spearphishing Attachment continues to be the most utilized sub-technique by attackers.

Microsoft Disabling Macros by Default

In February 2022, Microsoft announced that Excel 4.0 (XLM) macros will be disabled by default, in order to protect accounts from downloading malicious documents to their devices.

According to this radical change in the way Office documents are defined and can be used for malicious purposes, we would expect to see a sharp decrease in the use of attachments alongside an increase in the use of other techniques, such as Spearphishing Link or Exploit Public-Facing Application, but hackers have adapted to events in the technological field, and introduced new techniques in the world of attack, described on the next page.

Which files replaced Office files in recent phishing campaigns?

01

ISO and ZIP Files

The most noticeable trend we have seen since macros disabling is the use of ISO and ZIP files as initial vectors for phishing attacks.
These files are commonly used for compressing and packaging large amounts of data.
One common approach of this attack method is to include a malicious executable file within the compressed archive that is executed once the user extracts the contents of the attached file.
This executable may be designed to perform various malicious activities, install backdoors or persistent malware, manipulate system settings, and gather information.

02

HTML Files

The use of HTML files is not new for attackers, but has certainly increased in popularity since Office files are no longer a promising option.
HTML attachments are typically designed to look like legitimate emails and may contain malicious links or forms that encourage victims to click on them or enter sensitive information.
Attackers may embed malicious code or scripts within the HTML attachment to redirect the victims to a malicious website or download and execute malware onto their system.

03

OneNote Files

By the end of 2022, we identified a rising trend of multiple campaigns leveraging OneNote files to deliver malware payloads.
Those OneNote files are attached to malicious reply-chain phishing emails that hide within them embedded files that execute the initial payload of the malware. Another common technique used by attackers is creating a OneNote file attachment that contains a link or script that directs the user to a malicious website. This website might be a fake login page that tricks the victims into entering their login credentials to a site that lures them into downloading a malicious file.

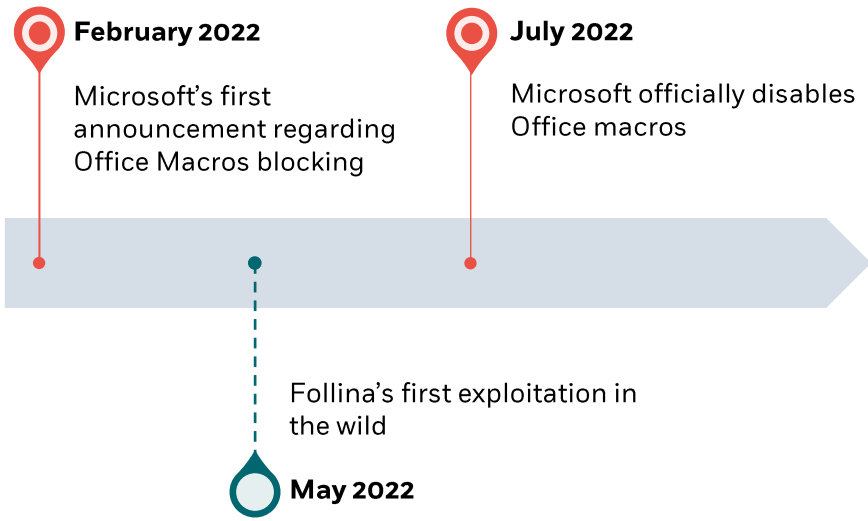
T1190 Exploit Public-Facing Applications

The CTI team identified that the second technique most utilized by attackers in 2022 was Exploiting Public-Facing Applications.

This figure is also consistent with the consequences of disabling Office macros, given that attackers had to find a way to bypass this configuration, and did so through a remote code execution (RCE) vulnerability in the Microsoft Support Diagnostics Tool (MSDT), or as it is officially called, Follina.

Once exploited, Follina enables attackers to load a malicious HTML file that permits the execution of a malicious PowerShell code into the compromised device.

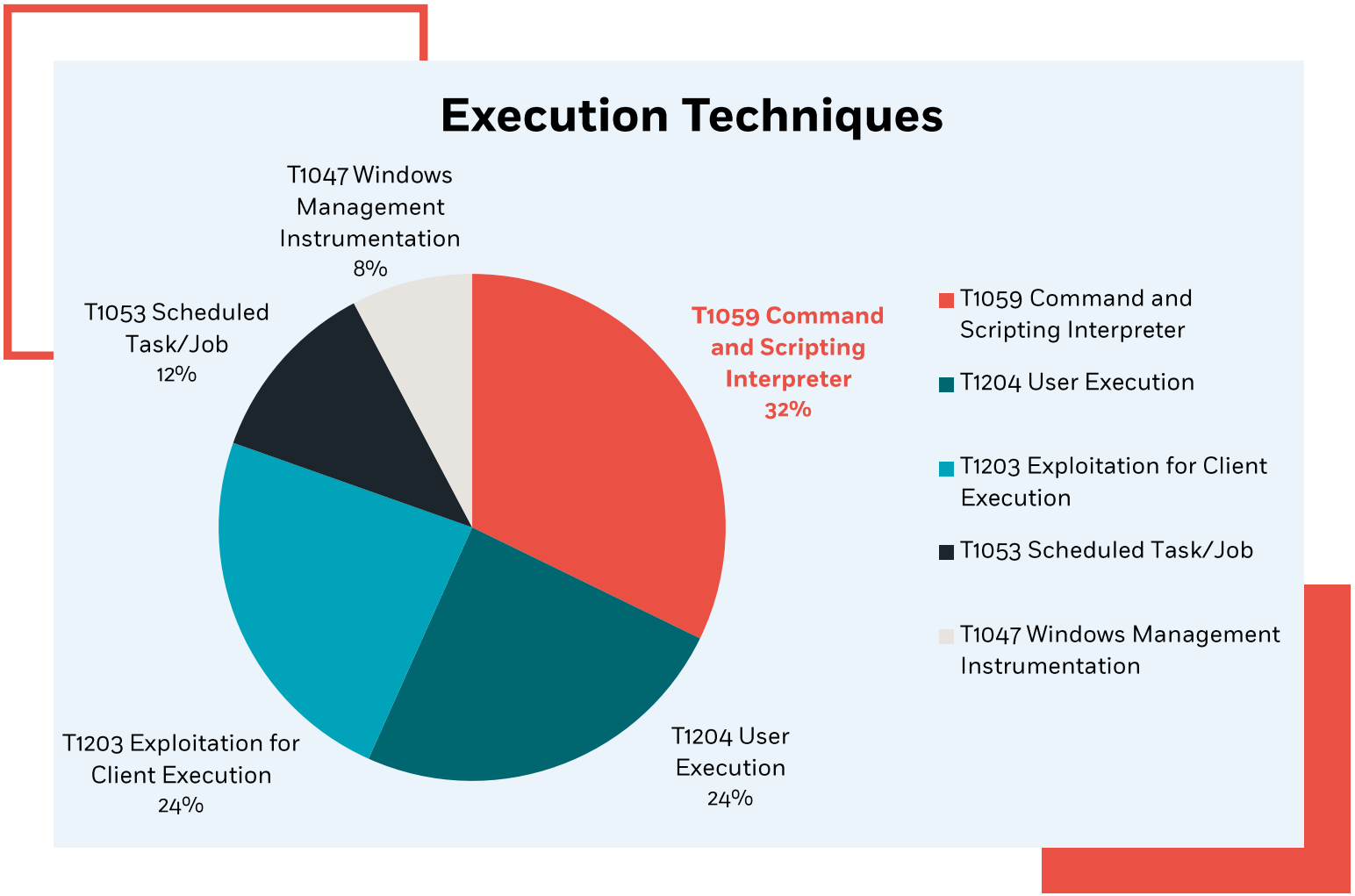
The exploitation of the Follina vulnerability for gaining access was used widely in the second half of 2022, following the "barriers" that were placed on attackers due to the macros disabling, affecting the amount this technique was used as an initial access vector.



CyberProof®
A UST Company

Tactic 2: Execution

Tactic 2: Execution – Technique Statistics



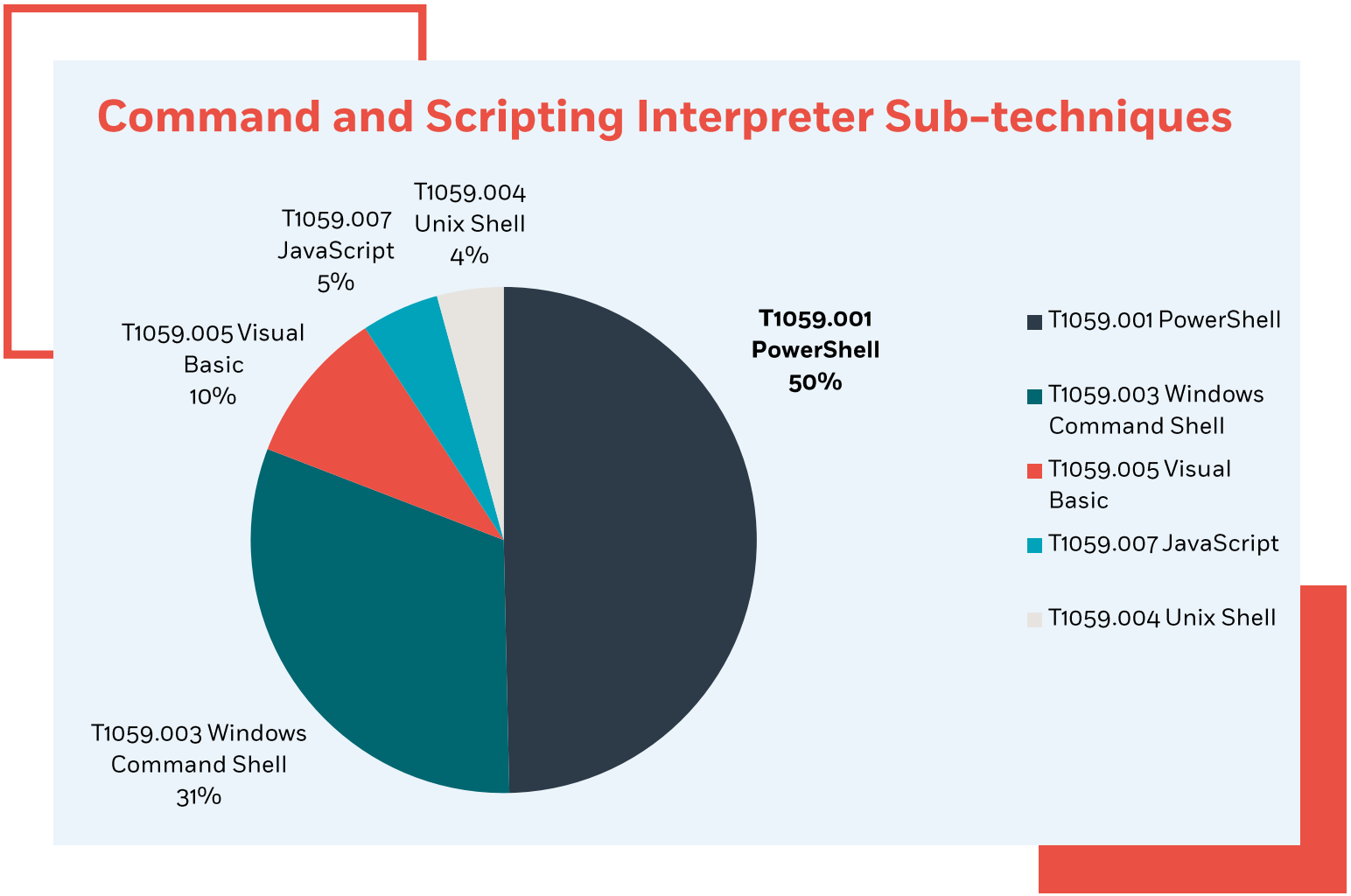
The leading technique used for Execution campaigns was the **T1059 Command and Scripting Interpreter**, as reported by the CyberProof CTI team.

T1059 Command and Scripting Interpreter

Command and Scripting Interpreter (CSI) is a software program that allows users to execute commands and scripts on a computer system. It provides a command-line interface that enables users to interact with the operating system directly.

In cyberattacks, CSI is widely abused to perform malicious actions such as malware and ransomware attacks. The popularity of CSI in cyberattacks can be attributed to its flexibility and customization capabilities.

Threat actors can easily create customized scripts and commands that can be executed on a compromised system, allowing them to steal sensitive data, alter system configurations, or install additional malware. CSI also provides remote access capabilities, enabling attackers to execute commands and scripts from a remote location, making it a valuable tool for targeted attacks.



By taking a deeper dive into Command and Scripting Interpreter Sub-techniques, we identified that **PowerShell** is the sub-technique most utilized by attackers, having been used in half of all cases.

T1059.001 PowerShell Sub-technique

PowerShell is a command-line interface and scripting language developed by Microsoft for Windows operating systems. It provides a powerful set of tools and commands that allows users to automate administrative tasks, manage system configurations, and execute scripts.

However, PowerShell has also become a popular target for threat actors and continues to be the most abused to execute attacks.

PowerShell is pre-installed on most Windows systems, making it easily accessible to attackers that can abuse it to execute malicious scripts and commands, evade detection, and maintain persistence in a compromised system.

Furthermore, PowerShell can be used to launch other attack techniques, such as **fileless malware**, which is difficult to detect and analyze.

What is Fileless malware?

Fileless malware represents an increasingly prevalent and sophisticated type of cyber threat that exploits legitimate system tools, such as PowerShell, to infect the victim's device and evade traditional security measures. In contrast to conventional malware that resides on a hard drive, fileless malware operates exclusively within a computer's memory, rendering it virtually invisible to antivirus software.

Why PowerShell?

PowerShell is particularly susceptible to fileless malware attacks due to its ubiquity, flexibility, and native access to system resources.

How PowerShell?

Threat actors capitalize on PowerShell's capabilities by crafting malicious scripts that manipulate its features to carry out nefarious activities, often without leaving a trace in the infected system. These scripts can be delivered via spear-phishing emails, malicious websites, or even social engineering tactics, which often lure victims into inadvertently executing the malicious code. Once executed, the malware can hijack PowerShell's legitimate functionalities to perform tasks such as data exfiltration, lateral movement within networks, and command-and-control communication, all while remaining concealed from traditional security tools.

Fileless malware attack flow[1]:

Stage 1:
The user gets a phishing email containing a malicious link

Stage 2:
The user clicks on the malicious link

Stage 3:
The malicious file loads flash on the victim's device

Stage 4:
The flash invokes the PowerShell tool

Stage 5:
PowerShell loads and execute a malicious script from a command and control server

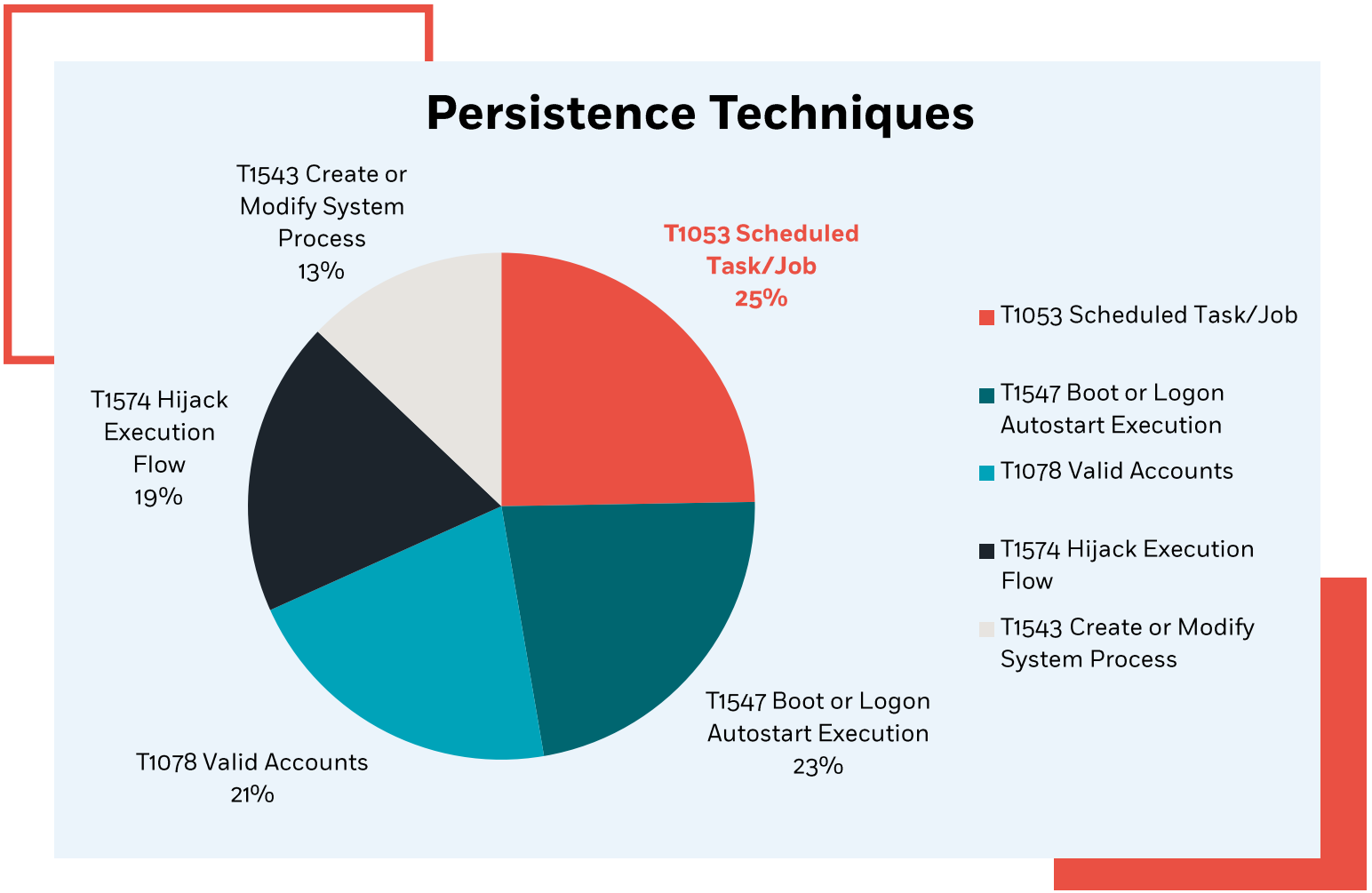
Stage 6:
A malicious code executes within the legitimate process memory

[1] CSO Online, Microsoft

CyberProof®
A UST Company

Tactic 3: Persistence

Tactic 3: Persistence – Technique Statistics



The leading technique used for Persistence in campaigns reported by the CyberProof CTI team was **Scheduled Task/Job**.

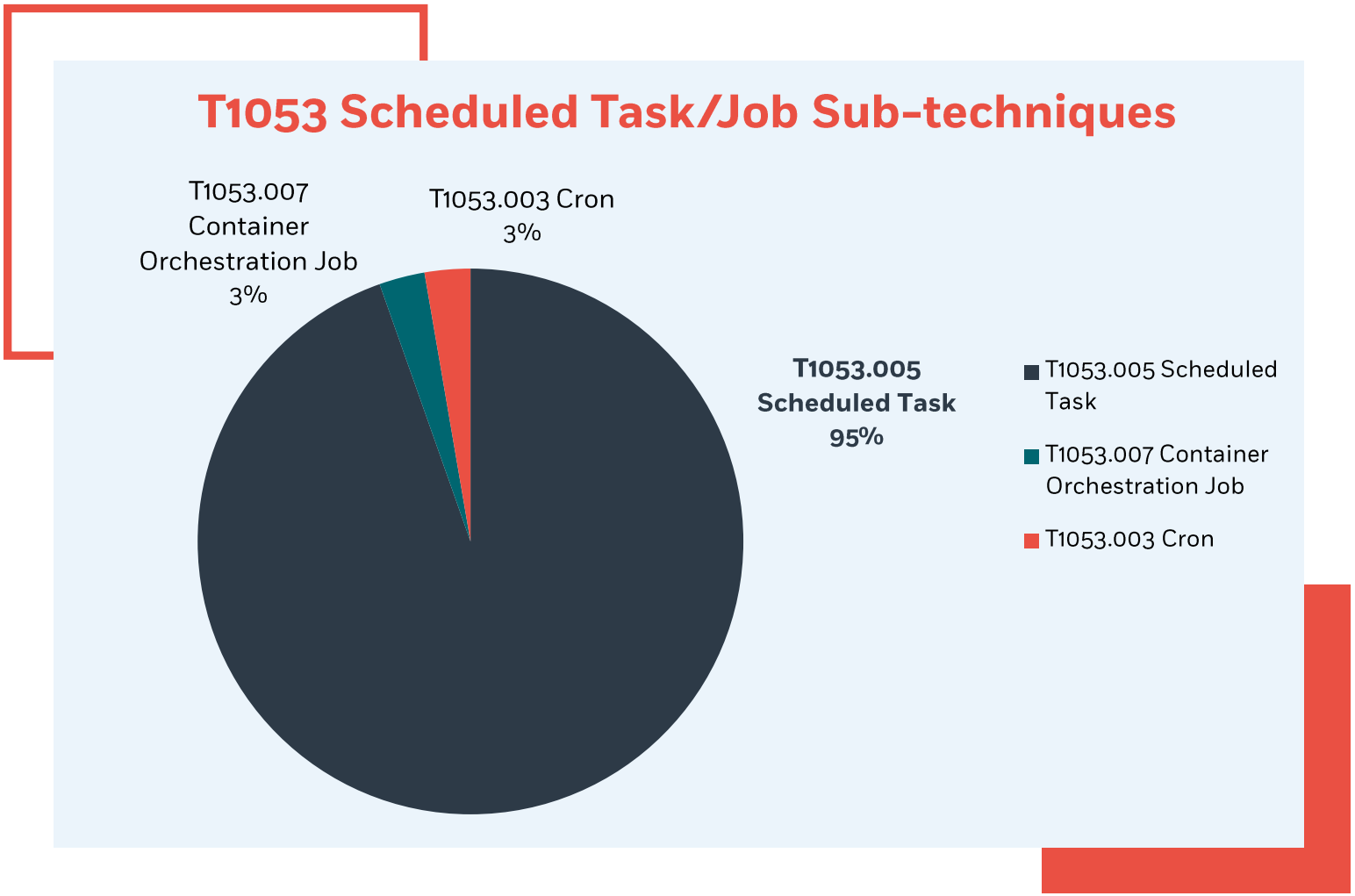
T1053 Scheduled Task/Job

Scheduled tasks are a built-in feature in operating systems that allow users to schedule programs or scripts to run automatically at specified times or events. They are commonly used for system maintenance, updates, backups, and other administrative tasks.

However, scheduled tasks have also become a popular target for threat actors, particularly in malware and ransomware attacks. Attackers can abuse scheduled tasks to execute malicious code or scripts, evade detection, and maintain persistence in a compromised system.

By creating a scheduled task that runs the malware at a specific time or event, attackers can easily bypass traditional security measures such as antivirus software. Moreover, scheduled tasks are often executed with system-level privileges, providing attackers with full access to the system and the ability to perform a wide range of malicious activities.

Due to their ease of use and versatility, scheduled tasks have become a common technique used by cybercriminals to launch and maintain their attacks.



By taking a deeper dive into Scheduled Task/Job Sub-techniques, we identified that **Scheduled Task** is the sub-technique most utilized by attackers in the past year, by an absolute majority over all the rest.

Tactic 3: Persistence – Sub-technique descriptions

A quick brief on T1053 Scheduled Task/Job sub techniques:

T1053.002 At	A command-line utility in **Linux and Unix** systems that allows users to schedule a **one-time task** to run at a specified time. It is commonly used for scheduling system backups, updates, or other administrative tasks.
T1053.003 Cron	A time-based job scheduler in **Linux** and **Unix** systems that allows users to schedule tasks to run **periodically**, such as daily, weekly, or monthly.
T1053.005 Scheduled Task	A utility in **Microsoft Windows** operating systems that allows users to schedule the **automatic execution** of programs or scripts at specific times or events. It is commonly used for system maintenance, updates, backups, and other administrative tasks.
T1053.006 Systemd Timers	A type of utility in **Linux** systems that enables users to schedule tasks to run at **specific intervals** or times. They can be used for scheduling system maintenance, updates, or other administrative tasks.
T1053.007 Container Orchestration Job	A utility used in **container-based** systems that manages the scheduling and execution of jobs or tasks within a containerized environment. It enables users to automate the deployment and scaling of containerized applications, ensuring that they run reliably and efficiently.

T1053.005 Scheduled Task

Although all the services mentioned above are capable of scheduling tasks, the Scheduled Task sub-technique has gained significant prevalence in the campaigns that we have examined, while the other utilities were either rarely encountered or absent entirely.
These figures immediately raise the question - what factors have contributed to the predominance of the Scheduled Task sub-technique?

01

Popularity and availability:

As Windows is the most widely used operating system, Windows Task Scheduler is more popular and well-known than the other utilities. Moreover, Scheduled Task is pre-installed on most Windows systems, making it easily accessible to attackers.

02

System-level access:

Windows Task Scheduler is executed with system-level privileges, providing attackers with full access to the system and the ability to perform a wide range of malicious activities.
In contrast, in Unix/Linux systems, these utilities are typically used by advanced users or system administrators who have elevated privileges.

03

Integration:

Windows Task Scheduler is integrated with other Windows utilities, such as PowerShell and Windows Management Instrumentation (WMI), which provides additional functionalities and ease of use. The other utilities may require additional software or tools to achieve the same level of integration.

04

Flexibility and Features:

Windows Task Scheduler provides a wide range of features, such as triggering tasks based on system events, user logins, or specific times. The other utilities have more limited functionality and are typically used for specific tasks, such as scheduling jobs or tasks in Unix or Linux environments. This versatility makes Windows Task Scheduler a more attractive target for attackers.

T1547 Boot or Logon Autostart Execution

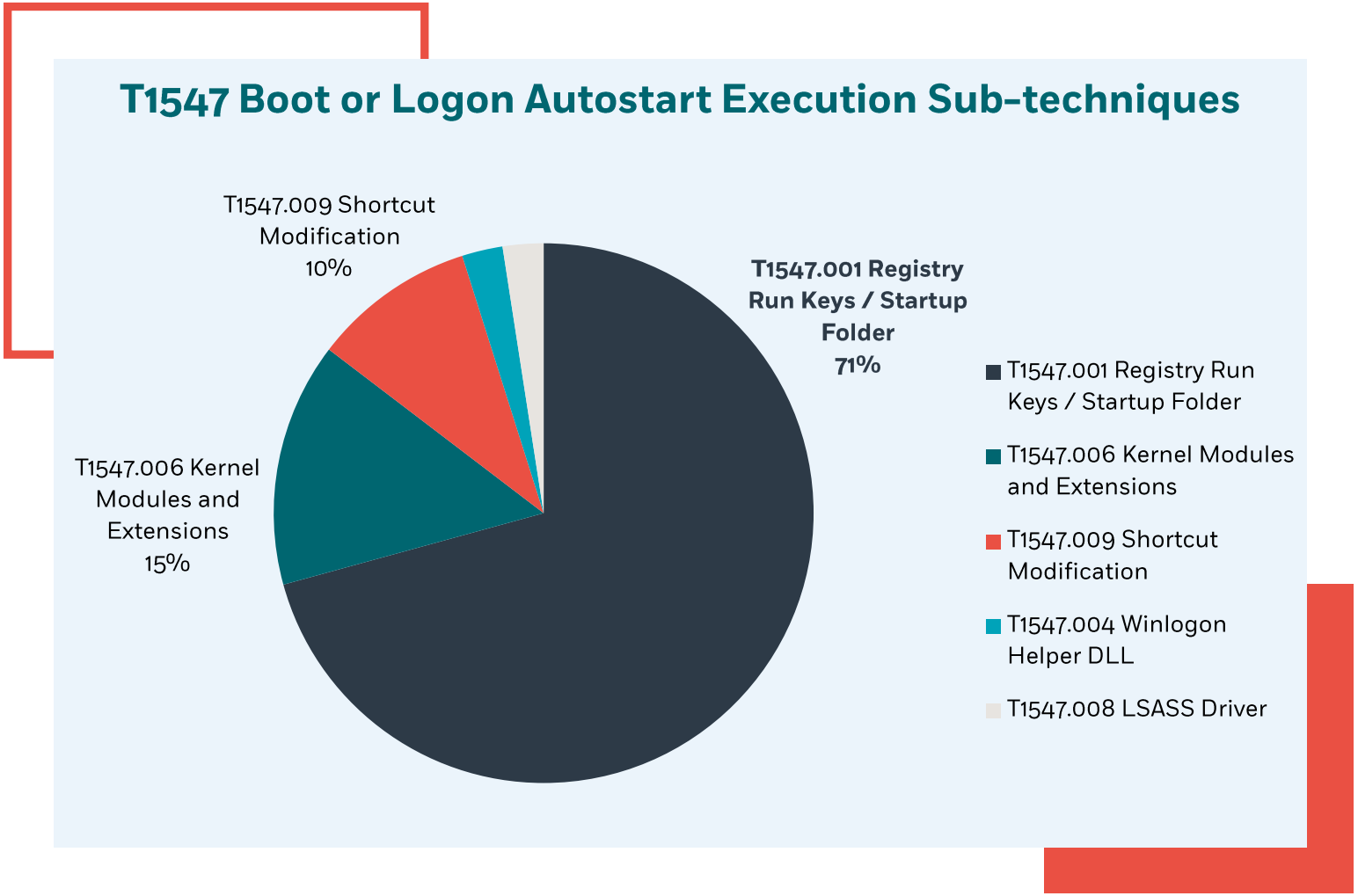
Boot or Logon Autostart Execution is a technique used by threat actors to launch malware automatically when a user logs into a compromised system.

This technique involves creating a registry key or modifying an existing one, enabling the malware to start automatically during the boot or login process.

Attackers use this technique to maintain persistence on a compromised system, ensuring that the malware runs every time the system is restarted or a user logs in.

This technique is popular among threat actors because it is easy to use and allows them to maintain control of the compromised system even after a reboot or shutdown.

Boot or Logon AutoStart Execution is often used in conjunction with other techniques such as scheduled tasks, making it harder for security systems to detect and remove the malware. Additionally, this technique is used to launch malware before any security system or antivirus solution loads, making it difficult to detect the malware's presence.



By taking a deeper dive into Boot or Logon Autostart Execution sub-techniques, we identified that **Registry Run Keys / Startup Folder** is the sub-technique most utilized by attackers in the past year, by an absolute majority over all the rest.

T1547.001 Registry Run Keys/Startup Folder

The **Registry Run Keys / Startup Folder** technique refers to the process of executing malicious code or programs during system startup by adding entries to the Windows Registry or Startup folder.

This technique is commonly used by threat actors to achieve persistence on a compromised system, as the malicious program will be launched every time the system boots up, allowing the attacker to maintain control of the system and carry out further malicious activities.

Abusing **Registry Run Keys / Startup Folder** is popular among threat actors because it allows them to maintain a persistent presence in the system even if the initial attack vector is remediated.

Additionally, this technique can be used to bypass security measures that may be in place, such as antivirus software or firewalls, as the malicious code is executed before these measures become active.

Moreover, threat actors can abuse this technique by adding entries to the Registry or Startup folder without the user's knowledge or consent, or by exploiting vulnerabilities in legitimate programs that allow them to add malicious entries. Once the malicious code is executed during startup, it can carry out a range of activities, such as stealing data, launching further attacks, or creating backdoors for remote access.

Tactic 3: Persistence – Technique 2: Boot or Logon Autostart Execution – Sub-technique 1: Registry Run keys/Startup Folder

Attackers may leverage various Registry Run Keys and Startup Folders to execute malicious payloads on targeted systems, according to the keys and folders' main purposes.

Some of the commonly targeted locations include the following[2]:

HKEY_LOCAL_MACHINE\Software\Microsoft\Windows\CurrentVersion\Run:
This is a Run key in the Windows Registry that contains a list of programs that automatically launch when the system starts up.

Who abuses this Run key? APT32

C:\Users%username%\AppData\Roaming\Microsoft\Windows\Start Menu\Programs\Startup:
This is a Startup Folder in the Windows file system that contains shortcuts to programs that automatically launch when the user logs in to the system.

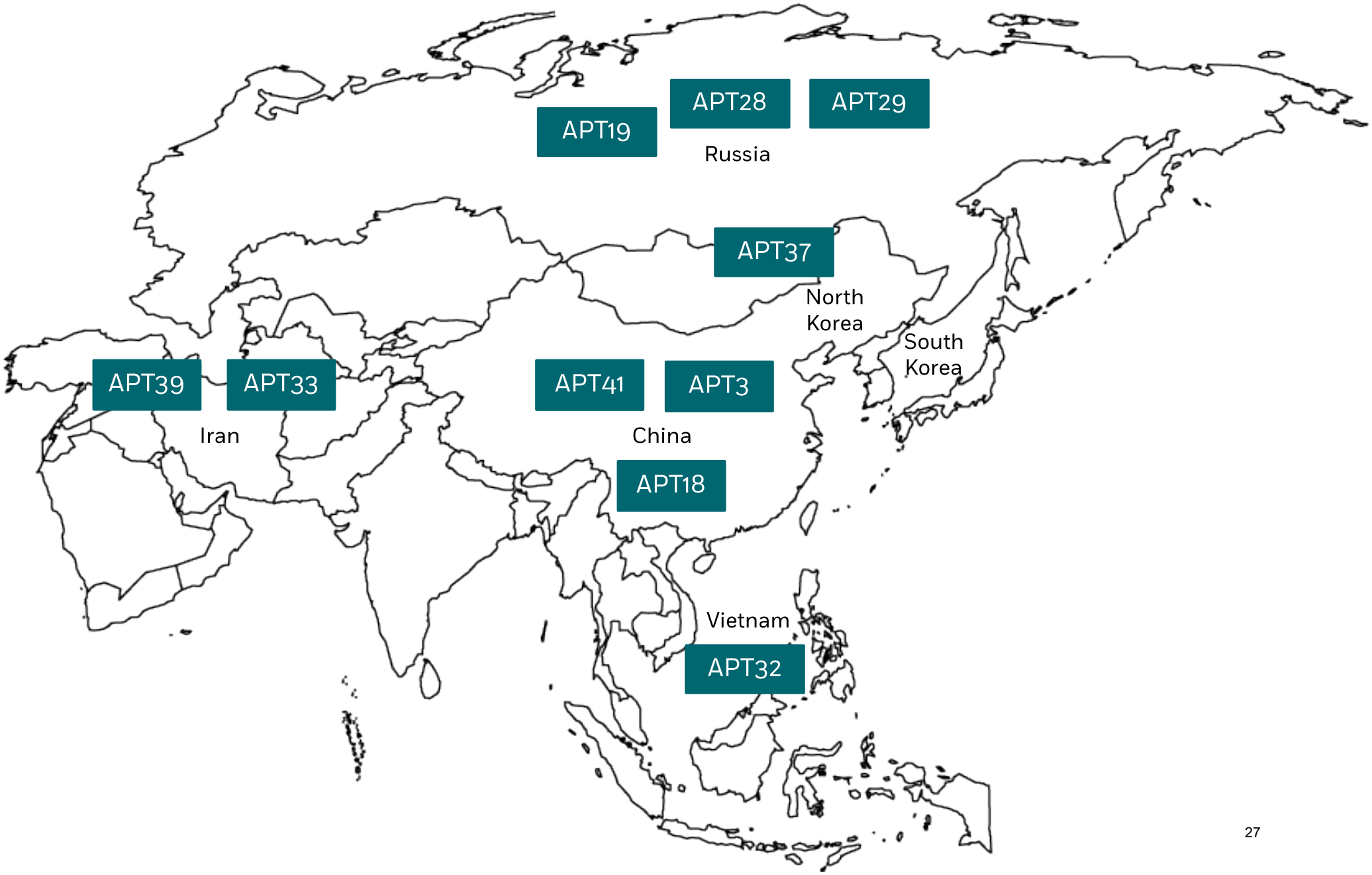
Who abuses this folder? APT28; APT33; APT41

HKEY_CURRENT_USER\Software\Microsoft\Windows\CurrentVersion\Run:
This is a Run key in the Windows Registry that contains a list of programs that automatically launch when a user logs in to the system.

Who abuses this Run key? APT18; APT19; APT29; APT3; APT32; APT33; APT37; APT39

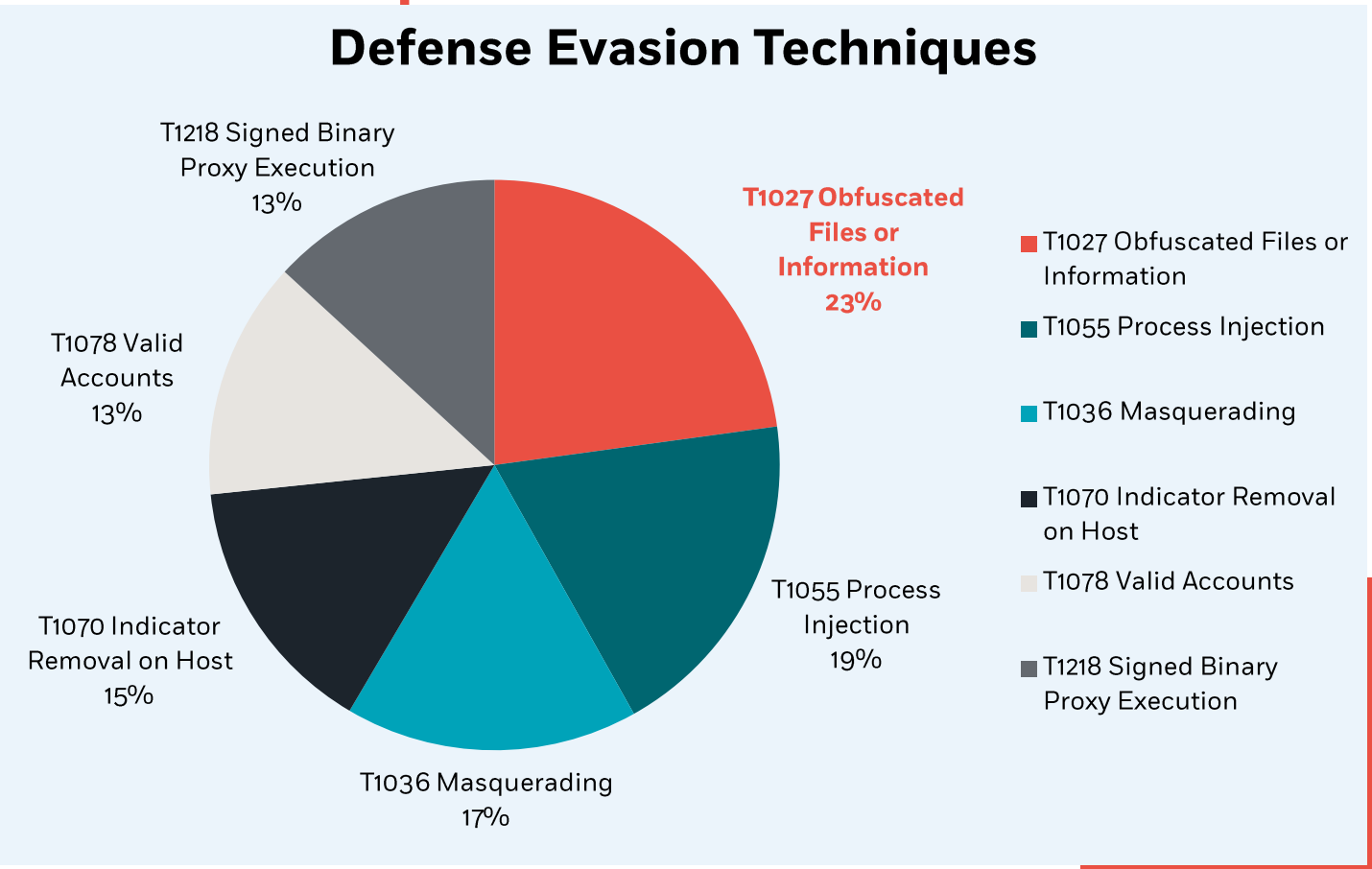
[2] MITRE ATT&CK Knowledge Base

APTs mentioned in this report



Tactic 4: Defense Evasion

Tactic 4: Defense Evasion – Technique Statistics



The leading technique used for Defense Evasion in campaigns reported by the CyberProof CTI team was **T1027 Obfuscated Files or Information**.

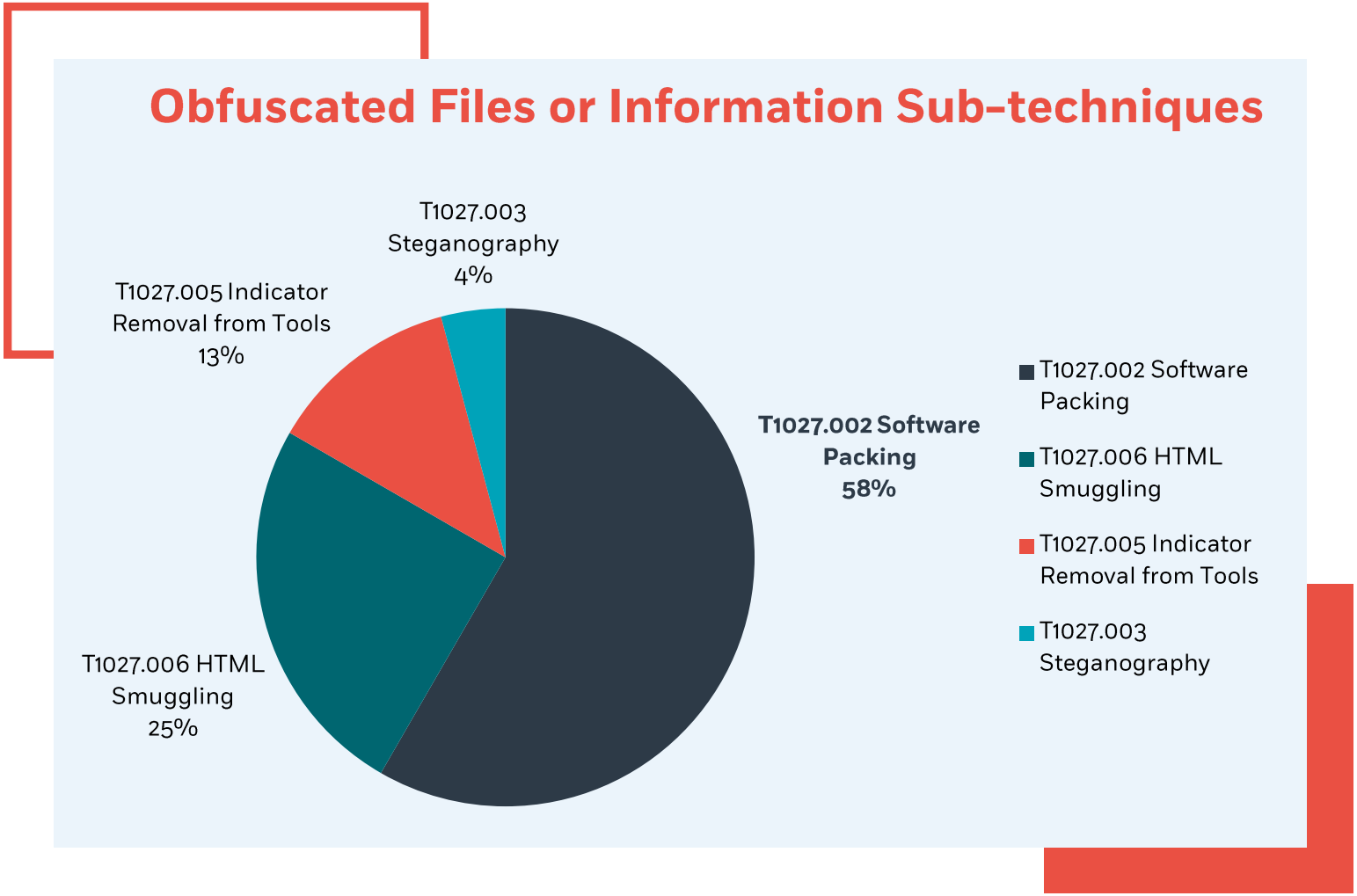
T1027 Obfuscated Files or Information

The technique of obfuscating files or information has become a mainstay among threat actors due to its efficacy in circumventing security measures.

By utilizing encryption, compression, or other methods to cloak the contents of a file or data, attackers can evade detection by security software and propagate malicious activities with impunity. This technique is particularly popular in malware and ransomware attacks, as it enables attackers to hide malicious code or sensitive data from traditional security systems, rendering them powerless in analyzing or intercepting the threat.

Obfuscated Files or Information also enable threat actors to bypass network security measures such as firewalls and intrusion detection systems, allowing them to exfiltrate data or communicate with command and control servers without fear of detection.

To successfully execute this technique, attackers often rely on social engineering tactics such as phishing to lure users into downloading or executing the obfuscated file.



By taking a deeper dive into Obfuscated Files or Information sub-techniques, we identified that **Software Packing** is the sub-technique most utilized by attackers, in an absolute majority over all the rest.

T1027.002 Software Packing

Software packing is a technique that is widely used by threat actors to obfuscate and conceal malicious code within legitimate software. This technique involves compressing and encrypting the malicious code, thereby making it difficult for traditional security tools to detect and analyze it.

The popularity of this technique can be attributed to its effectiveness in bypassing security measures and evading detection. Threat actors leverage software packing to hide the true intent of their malicious code, allowing them to carry out a wide range of nefarious activities such as stealing sensitive information, deploying malware, or gaining unauthorized access to systems.

How does the technique work?

Let's take the **AstraLocker Ransomware** campaign observed last year as an example:

AstraLocker 2.0 Uses Outdated Packer to Evade Detection

AstraLocker is a variant of the Babuk ransomware first detected in 2021. The Babuk ransomware was used by the same threat actor group, which operated a Ransomware-as-a-Service (RaaS) platform and licensed its software to affiliates to execute attacks.[3]

The AstraLocker 2.0, which was observed in March 2022, exhibits sophisticated tactics to evade analysis and detection, including the use of an outdated packer, the SafeEngine Shielden v2.4.0.0 protector, which employs indirect jumps every 5-7 instructions to obfuscate the control flow of the program.

This packer is so old that the legitimate version is no longer available, suggesting that the threat actor obtained a cracked version.

The packer also checks running processes and opens Windows names to detect if it is in an analysis environment and attempts to hide its threads from debuggers by setting the thread information argument HideFromDebugger.

3 Security Week News

T1027.006 HTML Smuggling

HTML smuggling is a stealthy technique used by threat actors to bypass network security measures and deliver malware to a victim's computer. This technique works by disguising malicious code within seemingly benign HTML code, which is then injected into a legitimate web page.

HTML smuggling is popular among threat actors because it allows them to bypass traditional security defenses, such as firewalls and intrusion detection systems, that are designed to detect and block malicious code.

Furthermore, HTML smuggling attacks can be carried out through standard web protocols, making them difficult to detect and trace. In addition, this technique enables attackers to evade content filtering technologies and deliver malware to unsuspecting users, thereby increasing the success rate of their attacks.

In attempts to find a replacement for Office file attachments following the macros disabling, attackers leveraged the use of HTML files, and as a result, increased the use of the HTML Smuggling technique, even within popular and well-known malware families, such as the Qbot malware.

How does the technique work?

Let's take the **Qbot** campaign observed last year as an example:

Qbot Campaign Leverages SVG HTML Smuggling to Infect Users

Qbot, also known as Qakbot, is a highly sophisticated and persistent banking Trojan that has been actively circulating since 2008. This malware is primarily designed to steal sensitive financial information from infected computers, including login credentials, banking details, and credit card information.[4]

In this Qbot campaign, the infection chain begins with a phishing email that has a malicious HTML file attached to it. As part of this campaign, phishing emails are sent as replies to existing email threads, usually old threads that were likely obtained and stolen by the Qbot operators in the past.

Furthermore, the HTML attachments contain base64-encoded Scalable Vector Graphics (SVG) images with smuggled JavaScript script. Once victims open the HTML attachments, the SVG images are decoded, resulting in the execution of the JavaScript script.

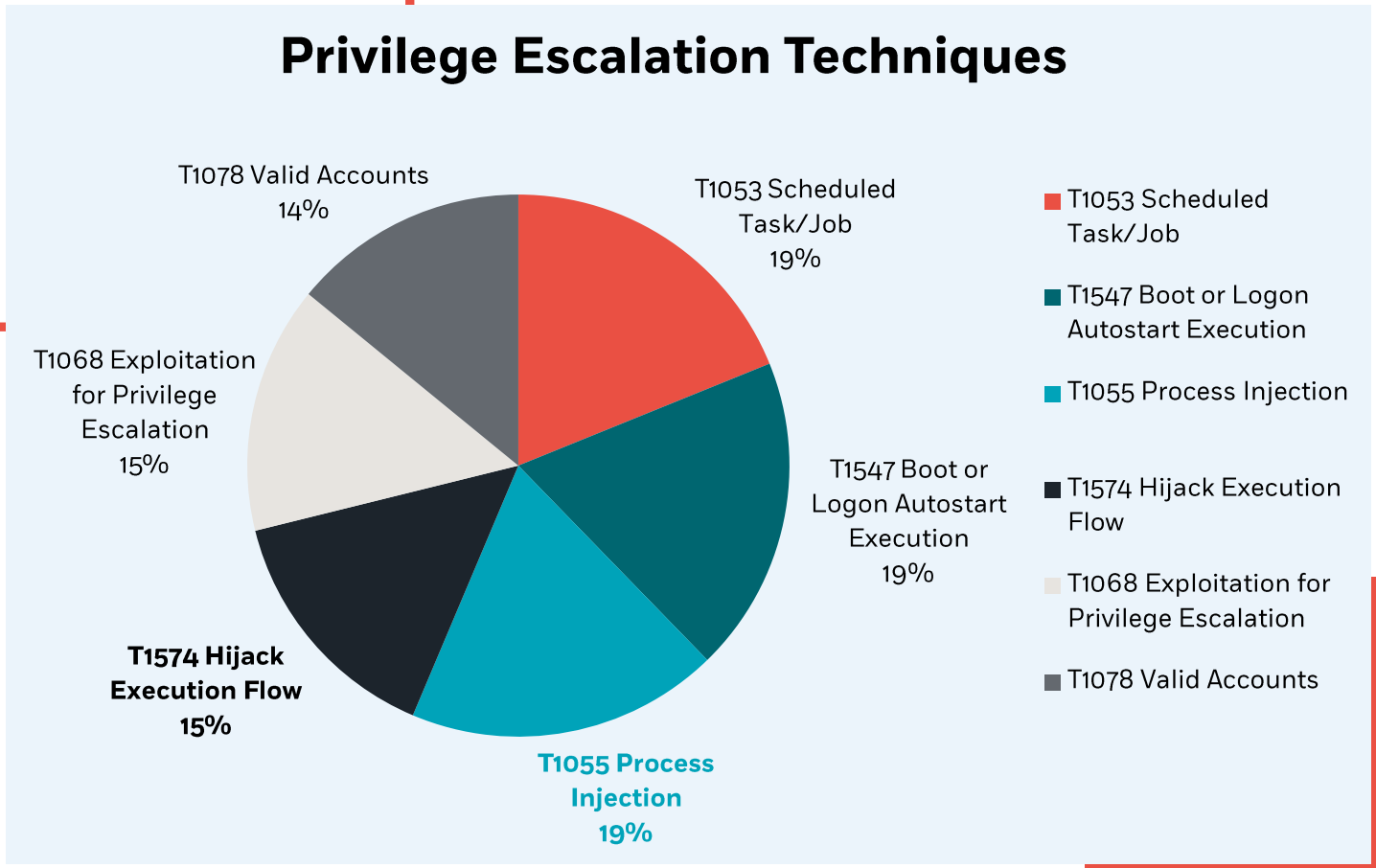
Following this execution, a malicious password-protected ZIP archive is created, prompting victims to save it on their machines. The password to the ZIP file is attached to the HTML file. Once saved, the archive extracts a malicious ISO image containing the actual Qbot payloads. In many cases, this multi-stage infection chain could result in a ransomware attack, as Qbot operates as a loader of other malware.

[4] The Hacker News

CyberProof®
A UST Company

Tactic 5: Privilege Escalation

Tactic 5: Privilege Escalation – Technique Statistics



One of the leading techniques used for Privilege Escalation in campaigns reported by the CyberProof CTI team was **Process Injection**, followed by **Hijack Execution Flow**.

T1055 Process Injection

Process injection is a technique that has become increasingly popular among threat actors in recent years. This technique involves injecting malicious code into a legitimate process running on a victim's system, allowing the attacker to evade detection and execute malicious actions without being detected by security systems. Process injection attacks can be carried out in various ways, such as DLL injection, code injection, and thread injection.

Threat actors leverage process injection techniques because they can evade traditional detection mechanisms that rely on file-based signatures or static analysis. By injecting malicious code into a legitimate process, threat actors can make it more difficult for security systems to detect and analyze their activities.

Additionally, process injection can enable threat actors to escalate privileges, steal sensitive data, or execute other malicious actions, making it a valuable tool for cybercriminals seeking to profit from their attacks.

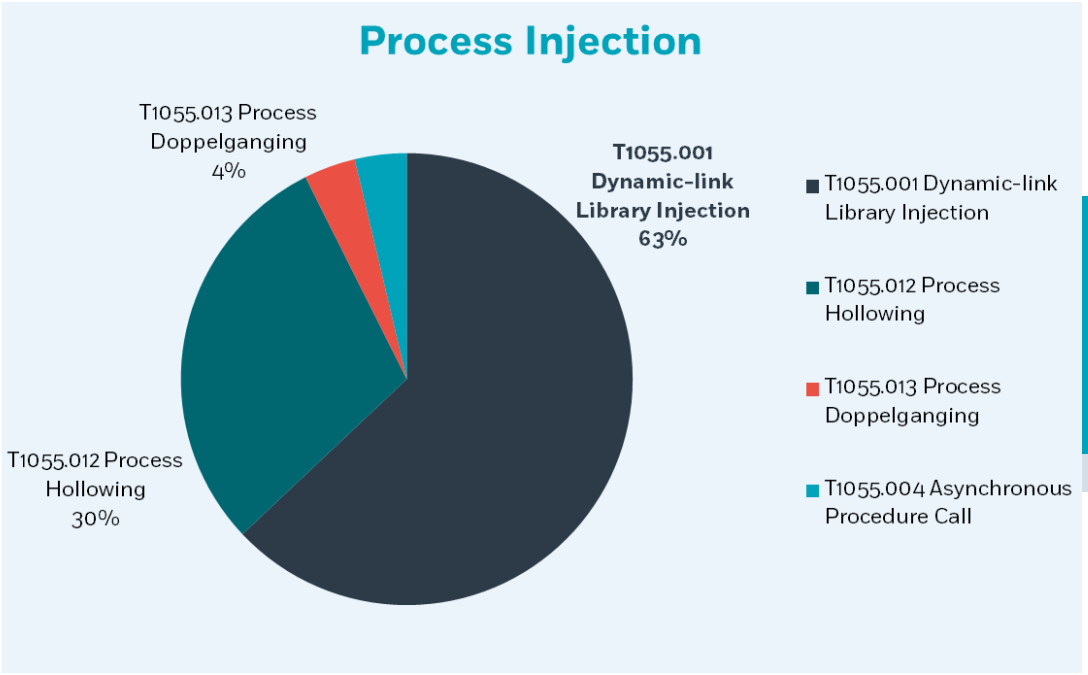
T1574 Hijack Execution Flow

The **Hijack Execution Flow** technique has become increasingly popular among threat actors seeking to escalate privileges and gain access to sensitive data.

Attackers leverage this technique to manipulate the flow of execution in a legitimate process by injecting malicious code or hijacking legitimate code paths. This allows them to execute malicious actions within the context of the legitimate process and evade detection.

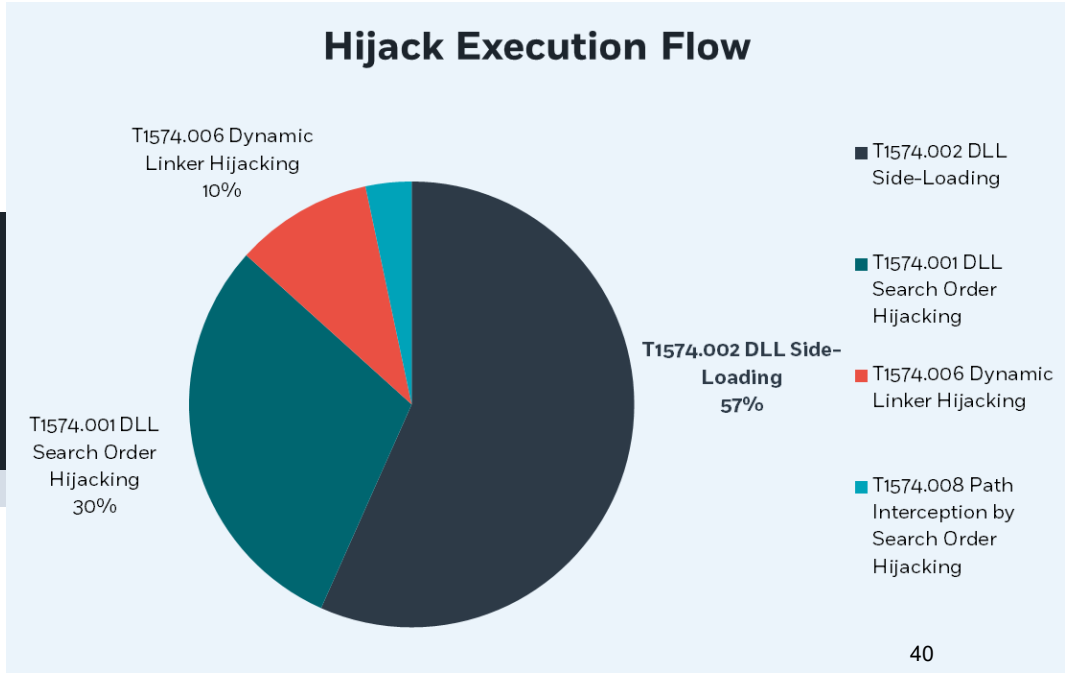
One of the key advantages of the **Hijack Execution Flow** technique is that it enables threat actors to bypass traditional detection mechanisms that rely on signature-based antivirus software or static analysis. By hijacking the execution flow of a legitimate process, attackers can evade detection and escalate privileges, giving them greater access to the victim's system and data.

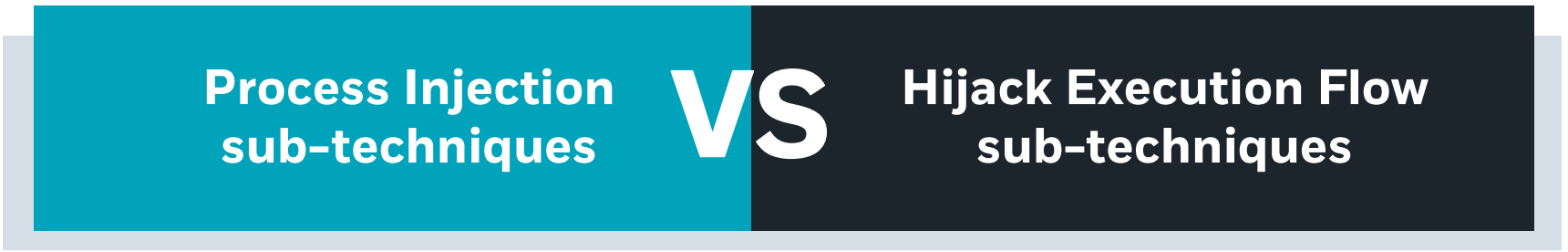
Tactic 5: Privilege Escalation – Technique 1 vs Technique 2: – Sub-technique Statistics



Process Injection sub-techniques

Hijack Execution Flow sub-techniques





What makes the difference?

DLL injection and DLL side-loading are two techniques used by attackers to load malicious code into a legitimate process. While both techniques achieve the same end goal, there are some key differences between them.

DLL injection involves injecting a malicious DLL file into a legitimate process's memory space, allowing the attacker to execute the malicious code within the context of the process. This technique can be used to evade detection and execute malicious actions without being detected by security solutions.

In contrast, DLL side-loading involves loading a legitimate DLL file into a process that does not call for it, allowing the attacker to execute malicious code through the legitimate DLL. This technique takes advantage of methods used by some applications to search for and load DLL files, which can be hijacked by attackers to load a malicious DLL file instead.

The main difference between DLL injection and DLL side-loading is the way that the malicious code is loaded into the legitimate process. With DLL injection, the malicious code is injected directly into the process's memory space, while with DLL side-loading, the malicious code is executed through a legitimate DLL that is loaded into the process.

T1068 Exploitation for Privilege Escalation

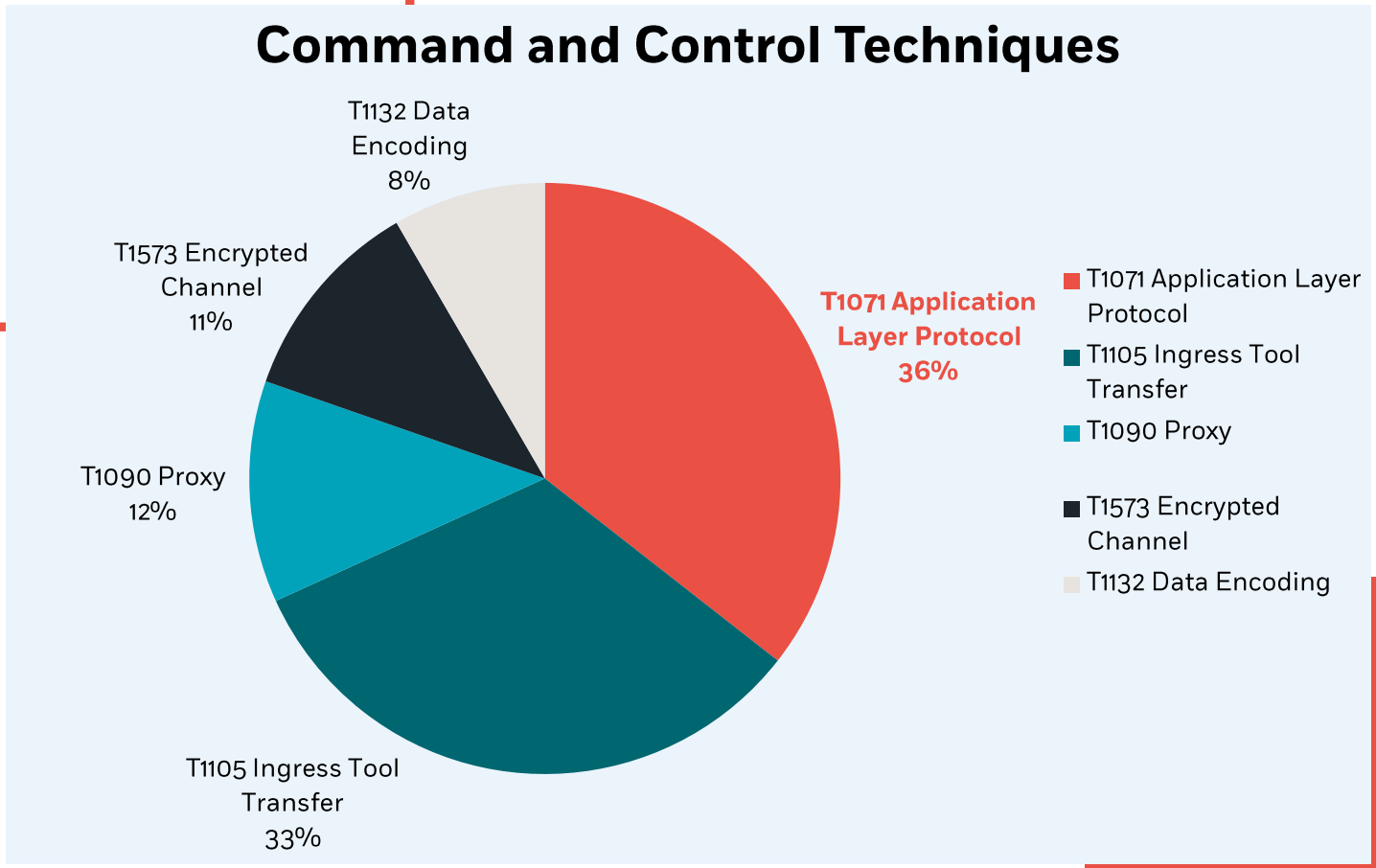
Exploitation for privilege escalation is a popular technique used by threat actors to elevate their level of access to a compromised system. This technique involves identifying vulnerabilities in the system, which can be exploited to gain higher privileges and access to sensitive data. Privilege escalation is a critical component of many cyberattacks, as it allows attackers to execute more advanced and damaging attacks on the compromised system. The popularity of this technique can be attributed to its effectiveness in enabling attackers to bypass security controls and gain privileged access to sensitive systems.

Some of the Privilege Escalation vulnerabilities we reported on during the past year:

CVE	Affected Product\Vendor	CVSS Score
CVE-2022-22947	Oracle Communications	10.0
CVE-2022-21431	Oracle Communications	10.0
CVE-2022-1388	BIG-IP iControl REST	9.8
CVE-2022-31137	Roxy-WI	9.8
CVE-2022-22955	VMware Workspace ONE Access	9.8
CVE-2022-22956	VMware Workspace ONE Access	9.8
CVE-2022-20700	Cisco Small Business Routers	9.8
CVE-2022-23131	Zabbix Frontend	9.8
CVE-2022-0811	CRI-O Container	8.8
CVE-2022-41040	ProxyShell	8.8
CVE-2022-0070	AWS's Log4Shell	8.8
CVE-2022-0185	Linux kernel	8.4

Tactic 6: Command and Control

Tactic 6: Command and Control – Technique Statistics



The leading technique used for Command and Control in campaigns reported by the CyberProof CTI team was **T1071 Application Layer Protocol**.

T1071 Application Layer Protocol

The Application Layer Protocol (ALP) is a networking technique used to facilitate communication between applications running on different systems. It operates at the highest layer of the OSI model and serves as an interface between the user and the underlying network infrastructure. ALP has gained popularity among threat actors due to its ability to exploit vulnerabilities in applications and systems. Threat actors can leverage ALP to bypass security measures, gain unauthorized access to systems, and steal sensitive data.

ALP-based attacks typically involve manipulating the protocol data unit (PDU) sent between applications to compromise the target system. These attacks can take various forms, including buffer overflows, SQL injections, cross-site scripting (XSS), and man-in-the-middle (MITM) attacks. By exploiting vulnerabilities in the application layer, threat actors can easily evade network security measures such as firewalls and intrusion detection systems (IDS).

One of the reasons ALP is popular among threat actors is its widespread use in modern networking protocols such as HTTP, FTP, SMTP, and DNS. These protocols are the backbone of many applications and services, and as such, they are prime targets for threat actors. Moreover, ALP-based attacks are often difficult to detect, as they can be disguised as legitimate traffic and evade signature-based detection systems.

T1055 Ingress Tool Transfer

The Ingress Tool Transfer technique has gained considerable notoriety for its efficacy in facilitating the malicious uploading of malicious payloads into target systems.

The technique exploits the vulnerabilities inherent in legitimate file transfer and synchronization protocols, such as File Transfer Protocol (FTP) and Server Message Block (SMB), to transfer malicious tools and resources into a compromised network.

The technique's widespread appeal among threat actors emanates from its capacity to obfuscate the origin of an attack, thereby circumventing traditional security countermeasures and evading detection.

By capitalizing on legitimate protocols, threat actors employ Ingress Tool Transfer to conduct an array of cyberattacks, ranging from the exfiltration of sensitive data to the deployment of ransomware and advanced persistent threats.

Disabling macros is a common thread:

According to our internal data, we have seen an increase in the use of the Ingress Tool Transfer in this past year, and we linked this figure also to the macro disabling that occurred this year in Microsoft Office, presented in the first slides of the report.

01

Bypassing email security:

Many email security systems scan email attachments for malicious content. However, ZIP and ISO files can be used as containers to store malicious payloads while evading detection. Since these file formats are compressed or encrypted, they often bypass the scanning process of email security systems, allowing the phishing campaign to reach its intended target.

02

Perceived legitimacy:

ZIP and ISO files are widely used and recognized formats for data storage and distribution. As a result, potential victims may be more likely to trust and open these file formats, assuming they contain legitimate content. Threat actors take advantage of this trust to deliver their malicious payloads.

03

Ease of distribution:

Using ZIP and ISO files makes it easier for threat actors to distribute a variety of malicious payloads. These file formats can contain multiple files, allowing attackers to package different types of malware or tools together in a single phishing email.

04

Social engineering:

Phishing campaigns often rely on social engineering tactics to deceive victims. By using ZIP and ISO files, threat actors can create seemingly legitimate emails that appear to be from reputable sources, such as software updates, tax documents, or other official communications, enticing the victim to open and execute the malicious content.

The most common tool in this year's reported campaigns:

Cobalt Strike

Cobalt Strike is a commercial, multi-platform, and modular penetration testing tool that was designed to simulate advanced persistent threat (APT) attacks back in 2012, and has since become a popular tool among penetration testers, red teams, and threat actors.

Cobalt Strike provides a range of capabilities that allow users to conduct targeted attacks and compromise networks. It includes a graphical user interface that allows users to create and deploy custom malware payloads, conduct phishing campaigns, and manage command and control (C&C) servers. It also includes a range of post-exploitation features that allow users to conduct reconnaissance, move laterally within a compromised network, escalate privileges, and exfiltrate data.

One of the key features of **Cobalt Strike** is its ability to evade detection by traditional antivirus and intrusion detection systems. It does this by using advanced obfuscation techniques to make the malware payload more difficult to detect by signature-based detection methods. It also allows users to customize the payload to fit the specific target environment, which makes it even harder to detect.

While **Cobalt Strike** was initially designed as a penetration testing tool, it has since been adopted by threat actors and APT groups for use in their own malicious campaigns. As such, its use by attackers is illegal and can result in severe legal consequences.

What makes Cobalt Strike such a winning tool for threat actors?

It's easy to use:

Cobalt Strike has a user-friendly interface and provides a range of features that allow threat actors to easily create and deploy malware payloads, conduct phishing campaigns, and manage command and control (C&C) servers.

It's difficult to detect:

Cobalt Strike uses advanced evasion techniques that help it avoid detection by traditional antivirus and intrusion detection systems. For example, it can use various obfuscation techniques to make the payload more difficult to detect by signature-based detection methods.

It provides a wide range of capabilities:

Cobalt Strike includes a wide range of capabilities that allow threat actors to conduct reconnaissance, move laterally within a compromised network, escalate privileges, and exfiltrate data. This makes it a useful tool for threat actors who want to maintain persistent access to a compromised network.

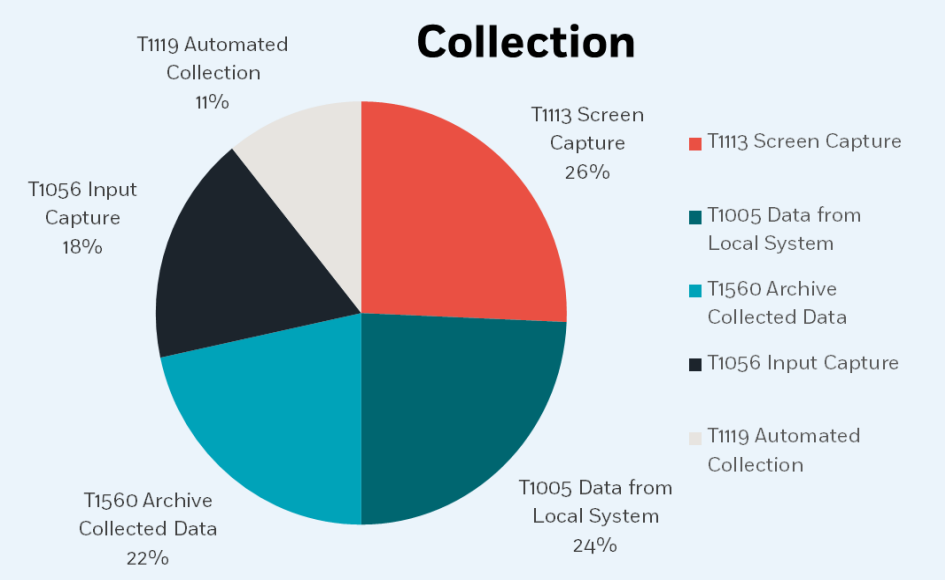
It's widely used and supported:

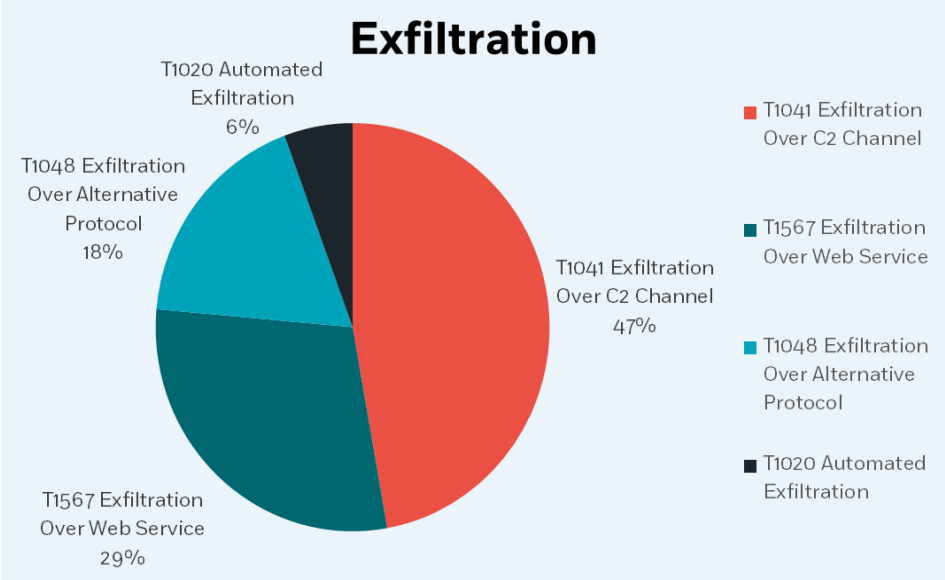
Cobalt Strike is widely used by penetration testers and red teams, which means that there are many resources available online to help threat actors learn how to use it effectively.

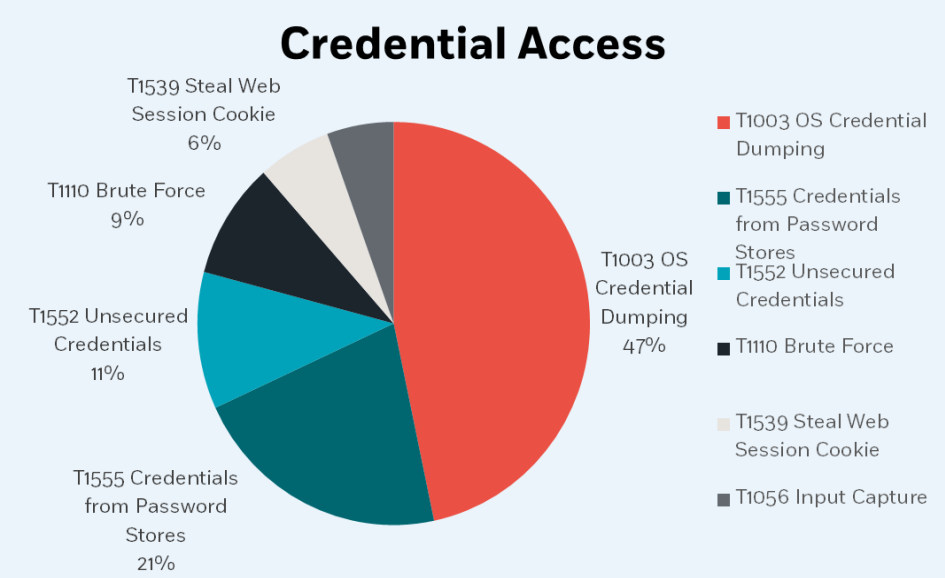
Additional MITRE Statistics

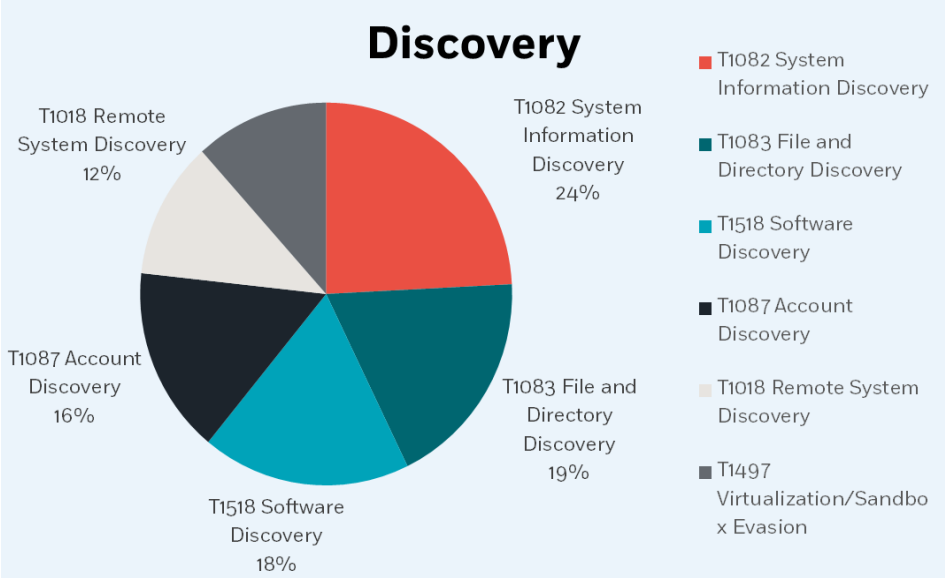
Statistics for MITRE ATT&CK techniques not covered in this report

The following charts show statistics for technologies related to additional MITRE tactics that are not covered in this report:









CyberProof®
A UST Company

Key Takeaways

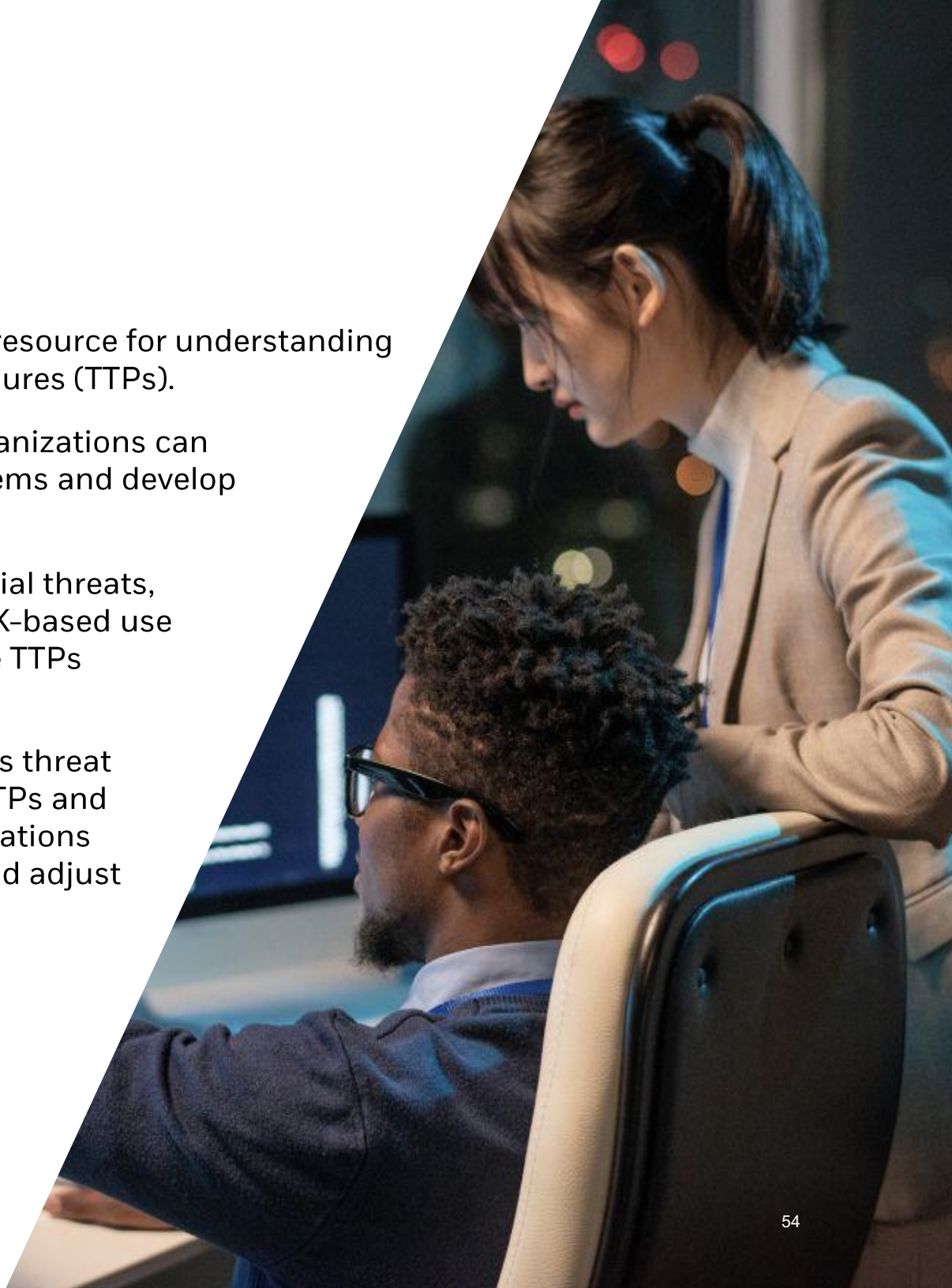
Key Takeaways

The MITRE ATT&CK framework is a valuable resource for understanding threat actors' tactics, techniques, and procedures (TTPs).

By analyzing the TTPs used by attackers, organizations can identify potential vulnerabilities in their systems and develop effective mitigation strategies.

To proactively identify and respond to potential threats, organizations should develop MITRE ATT&CK-based use cases that are tailored to their assets and the TTPs identified in their framework.

As we observed throughout the report, today’s threat landscape is constantly evolving, with new TTPs and attack vectors emerging all the time. Organizations must stay up-to-date on the latest threats and adjust their security measures accordingly.



CyberProof®
A UST Company

About CyberProof

CyberProof, a UST company, helps our clients transform their security to a cost-effective, cloud-native technology architecture. Our next-generation Managed Detection & Response (MDR) service is built to support large, complex enterprises by combining expert human and virtual analysts. Our services are enabled by our purpose-built platform, the CyberProof Defense Center – enabling us to be more agile, collaborate better, and deliver powerful analytics. Our integrated security services include Threat Intelligence, Threat Hunting, and Vulnerability Management. Our experts innovate to meet our clients' needs with custom use cases, integrations, and automations.

For more information, visit www.cyberproof.com.

Barcelona | California | London | Paris | Singapore | Tel Aviv | Trivandrum

cyberproof.com