2022

THREAT DETECTION REPORT

red canary

TABLE OF CONTENTS

2022

INTRODUCTION

Welcome to the 2022 Threat Detection Report

Welcome to Red Canary's 2022 Threat Detection Report. Based on in-depth analysis of over 30,000 confirmed threats detected across our customers' environments, this research arms security leaders and their teams with actionable insight into the threats we observe, techniques adversaries most commonly leverage, and trends that help you understand what is changing and why. This is our most expansive report to date, but our intention remains the same: The Threat Detection Report exists to help you understand and detect threats.

How to use the report:

- Start perusing the most prevalent **techniques**, **trends**, and **threats** to see what we've observed in our customers' environments.
- Explore how to detect, mitigate, and simulate specific threats and techniques.
- Talk with your team about how the ideas, recommendations, and priorities map to your security controls and your overall strategy.

New trends section

Red Canary's security operations team performs three primary activities:

- Our **Intelligence team** seeks to identify and understand distinct threats.
- Our **Detection Enablement** and **Detection Engineering** teams seek to understand these threats and engineer solutions that reliably detect them and enable timely investigation.
- Our **Incident Handling** team is charged with responding to threats before they harm customers.

In each of these areas, we've identified trends that help us understand how threats are evolving and how we as defenders must evolve in kind. From the continued scourge of ransomware to high-impact vulnerabilities and supply chain attacks, this section synthesizes intelligence with insights from the front lines of threat detection and response.

ACKNOWLEDGEMENTS

Thanks to the 100+ security experts, writers, editors, designers, developers, and project managers who invested countless hours to produce this report. And a huge thanks to the **MITRE ATT&CK®** team, whose framework has helped the community take a giant leap forward in understanding and tracking adversary behaviors. Also a huge thanks to all the Canaries—past and present—who worked on the 2019, 2020, and 2021 Threat Detection Reports. The Threat Detection Report is iterative, and parts of the 2022 report are derived from previous years.

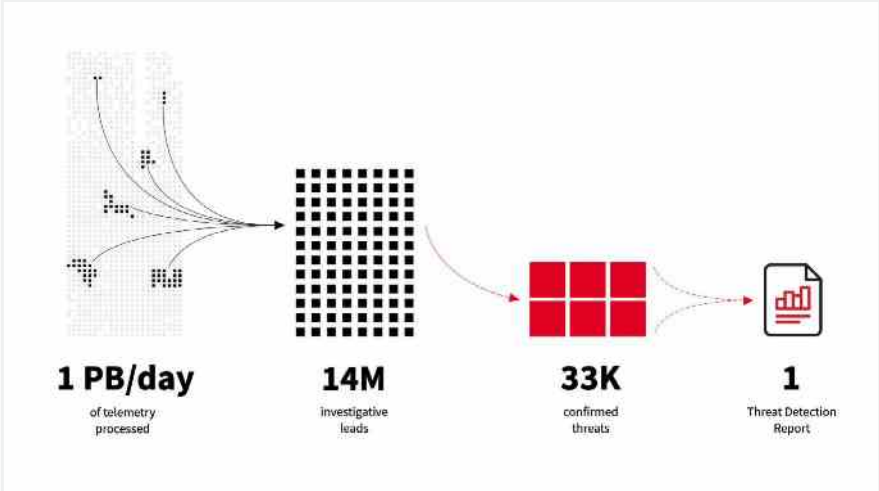
This report wouldn't be possible without all of you!

Methodology

Since 2013, Red Canary has delivered high-quality threat detection to organizations of all sizes. Our platform collects as much as a petabyte of security telemetry every day and leverages a library of roughly 3,000 detection analytics to surface potential threats that are analyzed by our Cyber Incident Response Team (CIRT). Confirmed threats are tied to corresponding **MITRE ATT&CK®** techniques and specific threats to help our customers clearly understand what is happening in their environments. A significant portion of this report is a summary of confirmed threats derived from this data.

Creating metrics around techniques and threats is a challenge for any organization. To help you better understand the data behind this report and to guide you as you create your own metrics, we wanted to share some details about our methodology.

Behind the data



To understand our data, you need to understand how we detect malicious and suspicious behavior in the first place. We gather telemetry from our customers' environments and feed it through a constantly evolving library of detection analytics. Our detection analytics are mapped to one or more ATT&CK techniques and sub-techniques, as appropriate. When telemetry matches the logic in one of our detection analytics, an event is generated for review by our detection engineers.

When a detection engineer determines that one or more events for a specific endpoint surpasses the threshold of suspicious or malicious behavior, they

create a confirmed threat documenting the activity on that endpoint. These confirmed threats inherit the ATT&CK techniques that were mapped to the analytics that first alerted us to the malicious or suspicious behaviors.

It's important to understand that the techniques and sub-techniques we're counting are based on our analytics—and not on the individual review performed by our detection engineers, during which they add more context to detections. We've chosen this approach to maximize efficiency and consistency. However, the limitation of this approach is that context gleaned during the investigation of a threat does not inform its technique mapping, and by extension, some small percentage of threats may be mapped incorrectly or incompletely. That said, we continually review these confirmed threats, and we do not believe that there are a significant number of mapping errors in our dataset.

How do you count?

You may be wondering how we tally the scores for the Threat Detection Report. Our methodology for counting technique prevalence has largely remained consistent since our first Threat Detection Report in 2019. For each malicious or suspicious detection we published during the year, we incremented the count for each technique reflected by a detection analytic that contributed to that detection (excluding data from detections of unwanted software). If that detection was remediated and the host was reinfected at a later date, a new detection would be created, thus incrementing the counts again. While this method of counting tends to overemphasize techniques that get reused across multiple hosts in a single environment (such as when a laterally moving adversary generates multiple detections within a single environment), this gives appropriate weight to the techniques you are most likely to encounter as a defender.

For the purposes of this report, we decided to set our rankings based on techniques, even though the majority of our analysis and detection guidance will be based on sub-techniques. This seemed to be the most reasonable approach, considering the following:

- Sometimes we map to a technique that doesn't have sub-techniques
- Sometimes we map to sub-techniques
- Sometimes we map generally to a technique but not to its sub-techniques

In cases where a parent technique has no subs or subs that we don't map to, we will analyze the parent technique on its own and provide appropriate detection guidance. However, in cases where sub-technique detections are rampant for a given parent technique, we will focus our analysis and detection guidance

entirely on sub-techniques that meet our requirements for minimum detection volume. To that point, we decided to analyze sub-techniques that represented at least 20 percent of the total detection volume for a given technique. If no sub-technique reached the 20 percent mark, then we analyzed the parent.

What about threats?

Our Intelligence team seeks to provide additional context about threats to help improve decision-making. By understanding what threats are present in a detection, customers can better understand how they should respond. Throughout 2021, the Intelligence team sought to improve how we identified and associated threats in detections. We chose to define "threats" broadly as malware, tools, threat groups, or activity clusters. We took two main approaches to associating a detection to a threat: automatically associating them based on patterns identified for each specific threat and manually associating them based on analysts' assessments conducted while reviewing each detection.

In contrast to our technique methodology, we counted threats by the unique environments affected. Whereas for techniques we counted multiple detections within the same customer environment as distinct tallies, for threats we decided to only count by the number of customers who encountered that threat during 2021. This is due to the heavy skew introduced by incident response engagements for laterally moving threats that affect nearly every endpoint in an environment (think ransomware).

Had we counted threats by individual detections, ransomware—and the laterally moving threats that lead up to it (e.g., Cobalt Strike)—would have been disproportionately represented in our data. We believe our approach to counting gives an appropriate measure of how likely each threat is to affect any given organization, absent more specific threat modeling details for that organization. It also serves as a check against the acknowledged bias in the way we count technique prevalence.

Limitations

There are a few limitations to our methodology for counting threats, as there are for any approach. Due to the nature of our visibility (i.e., that we predominantly leverage **endpoint detection and response data**), our perspective tends to weigh more heavily on threats that made it through the external defenses—such as email and firewall gateways—and were able to gain some level of execution on victim machines. As such, our results are likely different than what you may see from other vendors focused more on network or email-based detection.

While the top threats are worth focusing on, they are not the only threats to consider, since other impactful ones may be unidentified and therefore

underreported. The analysis and detection guidance in this report is reflective of the overall landscape, and, if implemented, offers a great deal of defense-in-depth against the threats that most organizations are likely to encounter. Knowing the limitations of any methodology is important as you determine what threats your team should focus on. While we hope our top 10 threats and detection opportunities help you and your team prioritize, we recommend building your own threat model by comparing the top threats we share in our report with what other teams publish and what you observe in your own environment.


2022
TRENDS

Trends

Red Canary performed an analysis of emerging and significant trends that we've encountered in confirmed threats, intelligence reporting, and elsewhere over the past year. We've compiled the most prominent trends of 2021 in this report to show major themes that may continue into 2022.

The **technique** and **threat** sections of this report are focused on detection data and identifying prevalent ATT&CK techniques and threats in those detections. The trends section takes us one step beyond that data and allows us to narrate events that might not be prevalent but may be emergent or otherwise deserve your attention.

How to use our analysis

The next page highlights the most prevalent threats occurring in our customer environments, so we can assume they are prevalent elsewhere. We include advice for responding to each threat and offer detection opportunities so you can better defend your organization. Some defenders may be able to take our detection guidance and apply it directly, while others may not. Regardless, defenders without a detection engineering function can still make use of the actionable analysis of each threat written by our Intelligence team experts.

Ransomware

Ransomware continued to dominate the 2021 threat landscape, and we observed operators take new approaches.

Supply chain compromises

Supply chain compromises were a major theme, starting with SolarWinds, Kaseya and NPM package compromises mid-year, and ending with Log4j.

Vulnerabilities

Adversaries exploited vulnerabilities affecting popular enterprise platforms to drop web shells, spread ransomware, and more.

Affiliates

The threat landscape continued its trend toward a software-as-a-service (SaaS) economy, muddying the already murky waters of attribution.

Crypters-as-a-service

Crypters like HCrypt and Snip3 joined the ranks of other "as-a-service" threats.

Common web shells

Adversaries exploited web applications with help from web shells such as China Chopper, Godzilla, and Behinder.

User-initiated initial access

We observed an uptick in threats that occurred after users sought out content which, often unbeknownst to them, was malicious.

Malicious macOS installers

Malicious installers led to rotten Apples and adware, as macOS systems continued to be targeted.

Remote monitoring and management abuse

Adversaries continued to use and abuse legitimate remote monitoring and management (RMM) software to move data and control infected hosts.

Linux coinminers

Coinminers once again dominated the Linux threat landscape.

Abusing remote procedure calls

Intrusions leveraging remote procedure calls (RPC) made waves, particularly PetitPotam and PrintNightmare.

Defense validation and testing

Confirmed testing comprised almost one quarter of our detections in 2021, with many coming from open source tools.

TREND

Ransomware

Ransomware continued to dominate the 2021 threat landscape, with operators taking new approaches.

Throughout 2021, ransomware remained one of the top threats to every organization. While some groups focused on traditional encryption, 2021 also marked the rise of additional tactics such as double extortion, which amplifies an adversary's leverage and further compels victims to pay up. Ransomware has become particularly challenging to track and prevent due to several trends we observed in 2021, discussed below.

The affiliate model

One challenge in responding to ransomware intrusions is that different adversaries are often involved at different phases of the intrusion. Ransomware groups usually rely on multiple affiliates to give them initial access to an environment before they encrypt files or take other actions. This makes tracking ransomware groups even more difficult, as intrusions can be a "mix and match" of different affiliates providing access to different ransomware groups.

Red Canary carefully tracks affiliates of ransomware groups and the malware they use, since these adversaries are the ones who sometimes gain initial access to an environment. These affiliates frequently use crimeware such as **Bazar** and **Qbot** to gain initial access to an environment, later passing off access to ransomware groups. A few common combinations of malware and ransomware we observed in 2021 include:

MALWARE FAMILY (PRECURSOR)	RANSOMWARE GROUP
Qbot	Egregor
Qbot	Sodinokibi/REvil
Qbot	Conti
Bazar	Conti
IcedID	Conti

Some things change, but some things stay the same

Challenges in understanding the ransomware landscape are not limited to tracking affiliates and payloads. Defenders must also contend with new groups emerging and others seemingly disappearing (often to be reincarnated in a different form as another group). Some of the ransomware families we bid farewell to in 2021 were Egregor, Sodinokibi/REvil, BlackMatter, and Doppelpaymer. While some seemed to fade away due to law enforcement actions, others disappeared for reasons that researchers haven't pinned down.

Where one ransomware family disappeared, however, another was ready to step into its place. 2021 saw the dawn of many new ransomware families, including **BlackByte**, **Grief**, Hive, Yanluowang, Vice Society, and CryptoLocker/Phoenix Locker. Many new ransomware families displayed close similarities to old families that "disappeared," leading analysts to assess that known adversaries simply resurfaced using a new name. For example, **Grief ransomware** displayed many similarities to Doppelpaymer, including its deployment following Dridex malware.

Beyond encryption

A significant ransomware trend in 2021 was the increase in adversaries expanding their threats beyond data encryption. Multiple ransomware groups pivoted to stealing and exfiltrating data before encrypting it, then demanding payment to prevent the data from leaking publicly on a dark web site. While this practice isn't new (it dates back to at least 2019), what was significant in 2021 was the number of groups who adopted this approach—to the point where it became the standard.

Adversaries realized they could demand payment for more than just the threat of a data leak or encryption. An adversary known as **Fancy Lazarus** (no affiliation with Fancy Bear or Lazarus Group) extorted victims by threatening to conduct a distributed denial of service (DDoS) intrusion if they didn't pay.

TAKE ACTION

There is no one simple way to prevent ransomware. The same security approaches you take to prevent any malware also should help prevent ransomware. It's critical to regularly update software, as we often see ransomware after operators exploit a vulnerability in an internet-facing application. Additionally, internet-facing remote desktop protocol (RDP) connections without multi-factor authentication (MFA) are a common ransomware vector, making MFA for any accounts that can log in via RDP a high priority.

Ransomware also frequently gets into an environment as a follow-on payload for malware delivered via phishing emails. Looking for these malware families, such as **Qbot**, **Bazar**, and **IcedID**, can be an effective way to identify a potential ransomware intrusion chain early and stop it in its tracks. Robust detection for other common post-exploitation behaviors and tools like **Cobalt Strike** are also effective in limiting the impact of ransomware, as adversaries conduct multiple phases before data exfiltration and encryption.

It's also important to remember that backups are no longer sufficient ransomware protection. While creating offline backups is an excellent security practice and may help restore an environment after a ransomware intrusion, organizations cannot rely on this entirely because adversaries regularly exfiltrate data before encryption, although this too offers potential opportunities for detection. Backups will allow an organization to get back up and running more easily, but will not protect you against leaked data.

While this report focuses on what security teams can do, when it comes to ransomware, it's also important to remember that this problem is monumental and extends beyond defenders. Policymakers are also taking a close look at ransomware, and it's necessary for the security community to help them better understand what we do so they can make better decisions.

TREND

Supply chain compromise

Supply chain compromises were a major theme in 2021, starting with SolarWinds, Kaseya and NPM package compromises mid-year, and ending with Log4j.

Supply chain compromises were prevalent in 2021, and these incidents aren't going away any time soon. It's important to understand the different types of supply chain compromises. To state it simply, a supply chain compromise occurs when an adversary compromises a software developer, hardware manufacturer, or service provider and uses that access to target customers who use the affected software, hardware, or service. For example, the SolarWinds and Kaseya incidents involved an adversary compromising update servers to target customers of the companies' IT management software. Separately, NPM package and Log4j incidents involved adversaries exploiting open source libraries in sweeping compromises that impacted products that use Log4j or NPM packages as a dependency—as well as anyone who uses those products directly. Each of these incidents made headlines in mainstream media as well as infosec publications.

SolarWinds

Adversaries compromised SolarWinds, accessed the update infrastructure for its Orion IT management software, and sent backdoored updates to the company's thousands of customers in December 2020, affecting organizations well into 2021. The trojanized Orion platform updates included a legitimately signed dynamic link library (DLL) file, and some featured backdoor functionality that, after a dormancy period lasting as long as two weeks, initiated communication with command and control (C2) servers. Adversaries identified targets of interest for further exploitation and conducted follow-on activity such as installing additional malicious binaries. These malicious binaries were used to install a backdoor where adversaries could access the victim organizations' accounts. SolarWinds had a massive impact across many networks, and it took months for enterprises to investigate and respond. This compromise initiated important discussions about supply chain risks that remain relevant in 2022 and beyond.

Kaseya

In July 2021, adversaries exploited vulnerabilities in Kaseya VSA IT Management software in a campaign ultimately designed to deploy Sodinokibi ransomware, also known as REvil. VSA is popular among managed service providers (MSP)

that use it to remotely administer IT systems. The adversaries exploited zero days to gain remote control over the MSPs' VSA installations, which they used to infect the MSPs' customers' endpoints with ransomware. Kaseya **estimated that about 50** direct customers who were running Kaseya VSA systems—and between 800 and 1,500 other businesses—were impacted by this breach. While the damage was not as bad as it **could have been**, this incident further highlights the importance of tracking supply chain threats. It also resulted in significant attention from the U.S. government, which later **indicted** the adversaries responsible for the intrusion. Read about **how Red Canary responded** to the compromises and protected several customers from ransomware infections.

NPM package compromises

Node Package Manager (NPM) is a repository for publishing node.js projects, including libraries that developers download and incorporate into their software to perform specific mathematical functions, process data in specific ways, visualize data, and more. In October 2021, adversaries compromised an open source JavaScript library with more than 7 million weekly downloads and used it to distribute password stealers and coinminers. At the time, the NPM registry did not require author accounts to use multifactor authentication (MFA), which led to an unknown adversary hijacking the registry accounts of multiple package authors. After hijacking, the adversary published malicious versions of the legitimate packages that contained malware. Victims included package authors and end users of applications relying on those packages. One package, **ua-parser-js**, was downloaded around 8 million times a week at the time and is used by Google, Amazon, Facebook, IBM, and Microsoft. The U.S. Cybersecurity and Infrastructure Agency (CISA) published a **security alert** about the incident, warning victims to update to a safe version.

There were many other NPM compromises throughout the year, most notably **us-parser-js**. **Prior to** this compromise, an adversary copied the legitimate **ua-parser-js** library and combined it with malicious code to publish a malicious library. **Following this compromise**, an adversary took control of two NPM packages, **coa** and **rc**. These incidents used a combination of XMRig coinminer on macOS and Danabot on Windows. Red Canary continues to track this activity.

Log4j

Log4j is a popular Java logging library underlying many third-party applications that was hit with a remote code execution vulnerability in December 2021. The primary threats initially exploiting this vulnerability were coinminers and botnets, though the community feared exploitation would expand because of Log4j's massive intrusion surface. In some scenarios, the Log4j library was affected by a remote code execution vulnerability.

One reason the community didn't observe a large volume of exploitation in the first few days may be that these vulnerabilities are highly application-specific, depending on how they've implemented Log4j. This means an adversary could not have crafted a single exploit that would have had a broad impact on many types of applications at once. Though it took adversaries a few weeks to ramp up targeting, in late December 2021 and early 2022, internet-facing VMware Horizon servers using vulnerable versions of Log4j became a target for multiple operators. Adversaries were likely attracted to VMware Horizon because it is widely used and often internet-facing. We anticipate the continued targeting of internet-facing applications using vulnerable versions of Log4j for months to come.

TAKE ACTION

One of the best ways organizations can be prepared is by accurately inventorying all of the hardware, software, and service providers they rely on and trust. This will assist in quick response when an inevitable supply chain concern arises. While it sounds commonplace, normal defense-in-depth strategies can also help prevent supply chain compromises from turning into impactful intrusions. Endpoint detection and response (EDR) tools coupled with network detection tools will help you detect malicious post-exploitation activity in the event an adversary gains access to your network through a trusted third party. While there may be nothing you can do to prevent a dependency or a vendor from being compromised, there is quite a lot you can do to detect and prevent follow-on compromise, including detection opportunities we've shared throughout the rest of this report.

TREND

Vulnerabilities

In 2021, adversaries exploited vulnerabilities affecting popular enterprise platforms to drop web shells, spread ransomware, and more.

Several high-profile vulnerabilities made it into the collective consciousness of the security community in 2021. ProxyLogon and ProxyShell targeted Microsoft Exchange servers and affected a massive number of systems, sometimes leading to ransomware deployment. The exploitation of vulnerabilities in Kaseya's VSA appliance software also led to ransomware deployment on some of the thousands of organizations that used Kaseya software for remote administration of endpoints. In the latter half of the year, adversaries exploited multiple vulnerabilities in Zoho's ManageEngine suite of products. PrintNightmare and an MSHTML vulnerability caused a ruckus among the security community and media; however, their actual impact appears to have been limited.

An important nuance to call out is that vulnerabilities are just flaws in code—a threat must exploit that vulnerability. Given the frequency with which vulnerabilities are disclosed and the ease with which adversaries can exploit newly reported weaknesses, particularly in common applications, Red Canary focuses on identifying and detecting the behavior we observe surrounding exploitation of a vulnerability. We recommend other organizations do the same. Understanding the threats and the ways in which adversaries operate in compromised networks allows defenders to protect against malicious activity regardless of the means by which their environment is accessed.

ProxyLogon (CVE-2021-26855, CVE-2021-26857, CVE-2021-26858, CVE-2021-27065)

In March 2021, Microsoft released details of **four Exchange Server vulnerabilities** collectively known as "ProxyLogon." If chained together, the vulnerabilities would allow an adversary remote code execution on a targeted Exchange server. Multiple adversaries, including the suspected Chinese state-sponsored group HAFNIUM, used the vulnerability chain to drop web shells and collect data from thousands of Exchange servers. Other adversaries used the **DearCry ransomware to target unpatched servers** as well. Microsoft released patches for these vulnerabilities at the time of initial reporting.

TAKE ACTION

We've outlined several of 2021's major vulnerabilities below, along with some detection guidance. Detecting exploitation of a vulnerability from an endpoint perspective can be difficult and depends on how exploitation occurs in practice. We have tried to supply detection guidance as close to the point of exploitation as possible. In other cases, we provide detection opportunities that would most likely appear as follow-on behavior, such as suspicious child processes or registry modifications. The targeting of vulnerabilities in enterprise applications and platforms is unlikely to slow down in 2022, so it's important to detect the threats that exploit them head-on.

Microsoft Exchange Mailbox Replication service writing Active Server pages

Adversaries exploited ProxyLogon to drop web shells on vulnerable systems, which manifested through the **msexchangemailboxreplication.exe** service writing an ASPX file to disk. Malicious web shells will likely be placed on the web server in a web-accessible directory. The following analytic looks for the Exchange mailbox replication service creating ASPX files.

process == **msexchangemailboxreplication.exe**

&&

filemod_extension == **.aspx**

ProxyShell (CVE-2021-31207, CVE-2021-34523, CVE-2021-34473)

Exchange servers remained a target throughout 2021. In July, Microsoft released details of three new vulnerabilities in the Exchange server, which were dubbed "ProxyShell." ProxyShell exploitation allows an adversary to **remotely execute code** without authentication. Following the exploitation, adversaries dropped web shells to conduct reconnaissance, move laterally, and in some instances, deploy **ransomware**. Where ProxyLogon seemed to have a high impact over a short period of time, ProxyShell seemed to persist throughout the year; we detected exploitation as late as December. DetectingProxyShell exploitation is similar to ProxyLogon mentioned above, specifically **msexchangemailboxreplication.exe** writing an ASPX web shell to disk.

PrintNightmare (CVE-2021-34527)

On July 1, security researchers and Microsoft released details of a new vulnerability dubbed "PrintNightmare" (CVE-2021-34527). PrintNightmare permits an unprivileged user to remotely obtain elevated privileges on any system running the print spooler service, which is enabled by default. It abuses a vulnerability in how the print spooler service fails to properly authenticate users attempting to load a printer driver dynamic link library (DLL). This zero day affected all editions of Windows, allowing code execution with local SYSTEM-level privileges.

Though the vulnerability was concerning, there were not many reported campaigns exploiting it. That said, ransomware operators such as **Vice Society** and **Magniber** have exploited the vulnerability to gain initial access, and therefore it's worth looking out for. We observed a single malicious instance of PrintNightmare exploitation leading to precursor ransomware behaviors.

Windows print spooler service spawning **cmd.exe**

PrintNightmare exploitation results in a shell being opened on the targeted system as a child process of the spooler service. This detection analytic identifies the Windows print spooler service spawning a shell on the system.

parent_process == **spoolsv.exe**

&&

process == **cmd.exe**

Kaseya VSA (CVE-2021-30116)

On July 2, adversaries leveraged multiple vulnerabilities in Kaseya Virtual Systems Administrator (VSA) to distribute Sodinokibi ransomware, also known as REvil. VSA allows IT administrators to remotely administer endpoints. By compromising this software, an adversary gains remote execution capability to a large subset of customer endpoints, especially if Kaseya is operated by a managed service provider (MSP).

Red Canary detected the initial behavioral activity using a preexisting analytic for identifying **certutil.exe** decoding content, as detailed below. Our Intelligence team had tracked Sodinokibi prior to this, which helped us identify the malicious registry modification of **blacklivesmatter** seen below and attribute it to Sodinokibi.

Certificate utility tool (**certutil.exe**) decoding content

This detection analytic will detect **certutil.exe** running with the **-decode** option. Adversaries frequently leverage certutil to decode Base 64-encoded content.

process == **certutil.exe**

&&

command_line_includes (**decode**)

ManageEngine products (CVE-2021-40539, CVE-2021-44077, CVE-2021-44515)

In November and December, we observed likely exploitation of remote code execution vulnerabilities in two different Zoho ManageEngine products: **ADSelfService Plus** (CVE-2021-40539) and **ServiceDesk Plus** (CVE-2021-44077). In one case, an incident response partner determined that ADSelfService Plus was used for initial access prior to deploying ransomware. The FBI noted that advanced adversaries exploited a vulnerability in a third ManageEngine product, **Desktop Central**. ManageEngine products are widely used among IT departments to manage various services across the enterprise. As such, this presents adversaries with a wide attack surface. Organizations using ManageEngine products in their environment should update accordingly. Patches for all the vulnerabilities listed here are available via **ManageEngine**.

Keytool.exe spawning system shell or PowerShell

For the vulnerability in ADSelfService Plus (CVE-2021-40539), we observed adversaries use the Java utility Keytool to move a web shell from the initial directory it was dropped into. As such, **keytool.exe** spawning shells should be investigated, and the following detection analytic should surface related activity.

```
parent_process == keytool.exe

&&

process == (cmd.exe || powershell.exe)
```

TREND

Affiliates

The threat landscape continued moving toward a software-as-a-service (SaaS) economy, muddying the already murky waters of attribution.

The term “affiliate” has been increasingly used to describe the cybercrime ecosystem’s evolution into a software-as-a-service (SaaS) economy. Borrowed from the subscription-based software specialization strategy, an “affiliate” refers to the provider-customer relationship of malicious services. In the cybercrime ecosystem, several SaaS variants have emerged, from **phishing-as-a-service** (PhaaS) to **access-as-a-service** to **crypter-as-a-service** to **ransomware-as-a-service** (Raas). It has never been easier to find an adversary for hire.

This service specialization across the phases of an intrusion has led to a proliferation of partnering, muddying the waters of what was once a relatively consistent collection of tactics across campaigns. As adversaries swap subscribers and pass off payloads, identifying and anticipating the progression of a compromise becomes more challenging. To meet this challenge, we need to distinguish the affiliate activity at each stage of the campaign.

Tracking threats at Red Canary

Tracking affiliates is tricky, and to help explain why we think it’s so important, we want to share some background on our threat tracking journey. At Red Canary, we primarily track threats by documenting their observable behaviors in the form of tactics, techniques and procedures (TTP). When we first set out on this intelligence mission, we began by clustering the most prominent and prevalent threats within our data. We often focused on the primary payload as a means of referring to the threat within a detection—think **Qbot**, **TrickBot**, or **Cobalt Strike**. Often we would see one or more of these threats progressing to another threat, especially in the wild west of active incident response engagements.

Throughout 2021, we realized that referring to activity as an Emotet phishing campaign or a Qbot phishing campaign was confusing. The activity we observed before and after Emotet or Qbot sometimes varied, while other times, we noticed the same patterns in how different malware families gained initial access. This realization helped us determine that patterns within filenames or infrastructure indicated that these characteristics likely belonged to their own initial access cluster—a delivery affiliate—rather than a simple malware payload as we had initially been referring to them. Understanding the relationships between these related threats enables us to better understand and respond to the overall ecosystem of the threat landscape.

Prominent affiliates in 2021

The process of teasing out the distinguishing characteristics that allow us to separate distinct clusters into more granular components is constantly evolving, as are the threats themselves. While we've been tracking some affiliates, such as **TA551** (named by Proofpoint), for quite some time, others came into focus more recently as our research progressed throughout the course of 2021. Breaking down intrusions into their component parts helps us better keep pace with the nature of the affiliate-based economy adversaries operate in today.

In 2021, we began identifying patterns in multiple phishing affiliates dropping variants of the **Bazar family** of malware, also referred to as "Baza." Derived from the use of **.bazar** top-level domains for C2 when it was first observed in the wild, this family has lent its name to multiple initial access vectors, campaigns, and components, including BazarLoader, BazarCall, and BazarISO. The multiple components under the umbrella of the Bazar family highlight the importance of differentiating the initial access from the payload. We have seen BazarBackdoor delivered by other prominent phishing affiliates, such as TA551, and have even seen behavior echoing some of the earliest campaigns that delivered BazarBackdoor surface in the latter half of 2021, delivering a resurgent Emotet as its payload.

Incorporating findings from other researchers helped us test hypotheses and add context to our understanding of several other affiliated threats. The prominence of Qbot in our detections and as a ransomware precursor led us to further scrutinize the XLSX phishing lures that delivered it. As a result of this research, we created a distinct profile for the TR delivery affiliate (which we also observed delivering IcedID). Distinguishing these components would not have been possible without other researchers who shared their findings, such as **Brad Duncan**.

Shifting away from phishing affiliates, we appreciated Morphisec's great reporting on **HCrypt** and **Snip3** in the first half of the year, the first time **crypter-as-a-service** crossed our radar. This helped us better break down several other clusters of activity to distinguish the hallmarks of the crypter from the initial phishing campaigns, such as Aggah, or the myriad RAT payloads HCrypt typically delivered.

TAKE ACTION

Analysts can better track affiliates by focusing on patterns in each phase of an intrusion and comparing similarities and differences to help distinguish when activity has passed from one affiliate to the next. To do this, you can ask questions of the data and compare answers across distinct incidents where you observed overlaps.
Here are some example questions to consider:

- Does the email that delivered this payload belong to a phishing affiliate, or is this entire campaign a cohesive cluster?
- What about the attachment or link within the email—is that a commodity maldoc? Is it part of access broker infrastructure, or does it belong to the adversary operating the later-stage payload?
- Is the download cradle and loader the beginning of the next-stage payload, or the last vestige of the delivery affiliate before handing off execution to the delivered payload?

By honing in on the handoff between one affiliate and the next, you gain better insight into the potential pivot points in the progression of an incident, hopefully detecting adversaries closer to the start of an intrusion. Distinguishing phishing affiliates such as TA551 or TR from the IcedID or Qbot payloads they deliver not only helps delineate the handoff between the affiliates, but allows you to dive deeper into delivery patterns to identify differences when the deployed payload changes. Anticipating the next stage of a threat's progression based on early observables enables defenders and incident responders to implement mitigations before that initial access can progress to lateral movement, data exfiltration, or ransomware.

TREND

Crypters-as-a-service

In 2021, crypters like HCrypt and Snip3 joined the ranks of other "as-a-service" threats.

Throughout 2021, Red Canary observed operators using crypters HCrypt and Snip3 to deliver various remote access trojans (RAT). Like other "as-a-service" threats, the developers sell or lease these crypters to affiliates who use them to carry out campaigns, expanding the threat landscape and creating new economies of scale. The "as-a-service" ecosystem lowers the technical barrier to entry, allowing operators to purchase capabilities rather than develop them.

HCrypt

HCrypt is a crypter designed to evade detection and facilitate the download of secondary payloads, often commodity RATs like ASyncRAT, Quasar RAT, and LimeRAT. We've seen adversaries leveraging HCrypt to gain initial access via phishing attachments, often relying on image files (IMG or ISO) containing a script (VBS or JavaScript) that launches HCrypt. The malicious script downloads an additional script hosted on various publicly accessible sites such as GitHub and Discord. Without intervention, this execution chain ultimately leads to a RAT infection.

Snip3

Like HCrypt, Snip3 is a crypter designed to evade detection and download additional malware. Snip3 is often delivered via phishing emails that prompt victims to download a VBA file. To evade detection, Snip3 leverages obfuscated PowerShell commands that contain the **RemoteSigned** flag. We've observed these PowerShell commands connecting to top4top[.]io, a legitimate file-sharing service popular in Egypt, Algeria, and Yemen.

Because these crypters are used by various adversaries delivering different payloads, it can be difficult to cluster seemingly disparate activity. However, as public reporting on Snip3 has discussed specific targeting of victims in the aviation sector, and we know of at least one set of operators that consistently relies on phishing emails with lures related to travel or cargo, we've associated activity we saw in 2021 with a campaign Cisco calls Operation Layover. We assess with high confidence that certain activity we observed in 2021 overlaps with this long-running operation, also chronicled by researchers from Morphisec and Microsoft. While this campaign involved attempts to deliver ASyncRAT or RevengeRAT to victims, similar intrusion chains deliver other publicly available RATs.

TAKE ACTION

As HCrypt and Snip3 operate "as-a-service," groups that purchase these capabilities may leverage them in different ways. The detection analytic below represents one opportunity to detect both crypters, empowering defenders to intervene before adversaries deliver additional malware.

Detection opportunities

WScript spawning Powershell using Invoke-Expression

This detection analytic will identify **wscript.exe** spawning PowerShell that uses Invoke-Expression or one of its aliases. HCrypt and Snip3 use PowerShell Invoke-Expression cmdlets to execute downloaded PowerShell content filelessly, without the downloaded scripts touching disk.

```
process == powershell.exe
&&
parent_process== wscript.exe
&&
command_line_includes  (iex || invoke || invoke-expression)
```

TREND

Common web shells

In 2021, adversaries exploited web applications with help from web shells such as China Chopper, Godzilla, and Behinder.

Web shells seriously affected many environments in 2021 due in large part to Microsoft Exchange and Zoho ManageEngine web server exploitation. Throughout the year, adversaries exploited ProxyShell, a Microsoft Exchange vulnerability, to gain privileged access to email systems owned by thousands of organizations. In these cases, the adversaries left behind a China Chopper web shell, a small and extensible bit of code that runs arbitrary ASP.NET, PHP, JSP, and other languages. Some versions of China Chopper require authentication with a preset password, but many adversaries fail to implement this, meaning that multiple adversaries can use the same web shell in different campaigns at once.

We also observed adversaries exploiting Oracle WebLogic servers to install the Godzilla web shell. Like China Chopper, Godzilla supports execution in ASP.NET, JSP, and PHP. Unlike China Chopper variants though, Godzilla web shells use a combination of simple password authentication with an additional encryption key value to require adversaries to have two pieces of information to communicate with the shell. The authentication ensures that only a single adversary can use the web shell. This encryption also obfuscates network traffic and confounds network-based analysis.

Finally, we observed the Behinder web shell following adversaries exploiting a Java-based web application made by Chinese cloud software company Yonyou. Behinder can load and execute compiled payloads in addition to standard commands. As with Godzilla, Behinder supports encryption and goes the extra mile by randomizing User-Agent strings in network traffic to hinder network and log analysis.

Why web shells matter

Web shells are malicious scripts designed to maintain persistent access to compromised web servers and facilitate remote code execution. Some are simple, allowing adversaries to issue a single command in a text box on a web page, while others include extensive capabilities where the adversary's imagination is the limit. Web shells execute with the same user account privileges as the exploited web application. If the application runs as an administrator, sensitive databases and systems may be accessible. Most web shells have simple or non-existent authentication mechanisms. Adversaries often leave web shells on public-facing web servers with no authentication mechanisms so they can return to the systems

later. In some incidents, responders may find many web shells on a single server or evidence of multiple adversaries using an abandoned web shell. Web shells should be removed as soon as possible to prevent further access.

TAKE ACTION

Patching should be the first step for remediating vulnerable web applications like **Exchange**, to prevent web shells from being dropped at all. Look for evidence of an existing breach by following guidance from the application developers. For example, **Microsoft recommends** using the Microsoft Support Emergency Response Tool (MSERT) to scan the Exchange server for exploitation.

If you cannot patch your web applications, consider creating IIS rewrite rules, disabling Unified Messaging services, and disabling multiple Internet Information Services (IIS) application pools. These stopgap measures may affect the internal and external availability of your applications, depending on which products your organization uses. For more remediation advice, check out our blog **Microsoft Exchange server exploitation: how to detect, mitigate, and stay calm**.

To detect web shells, start by examining file modifications and process executions. For Exchange servers, look for suspicious ASPX file modifications that may indicate an adversary wrote a web shell to disk. For other web applications like ASP.NET, PHP, and JSP applications, look for suspicious process behaviors. For example, you may be able to identify web shell activity by watching for web server worker processes spawning **cmd.exe** and PowerShell, **certutil** on Windows, or **curl** on Linux systems.

Windows IIS Worker process spawning **certutil.exe**

This detection analytic will identify unusual activity originating from **w3wp.exe** executing **certutil** to download files.

parent == **w3wp.exe**

&&

command_line_includes (**certutil** && **-split**)

Linux PHP or Java processes spawning **wget** or **curl**

This detection analytic will identify unusual activity originating from Linux web servers executing **wget** or **curl** to download files.

parent_process == (**php** || **java**)

&&

command_line_includes (**wget** || **curl**)

TREND

User-initiated initial access

We observed an uptick in threats that occurred after users sought out content which, often unbeknownst to them, was malicious.

In 2021, Red Canary observed adversaries use a range of initial access mechanisms to gain a foothold into victims' environments. Much of the activity we saw was consistent with our expectations, with many detections resulting from malicious emails, attempts to harvest victims' credentials, and breaches by way of a trusted party. Additional details on trends associated with these initial access vectors and follow-on activity such as webstall installation can be found throughout the report.

Understanding initial access can help defenders protect their environments early on. Prioritizing detections related to initial access saves money, time and effort; lessens pain points for users; and reduces impact to a business. From an intelligence perspective, understanding common patterns in initial access and follow-on activity helps build confidence in determining if relationships exist between threats that co-occur in an environment.

Notably, over the past year, we observed a rise in what we refer to as "user-initiated activity:" cases where victims downloaded a malicious executable after engaging with content they purposefully sought out. This often occurs without the victim's knowledge, particularly in cases where adversaries poison search engine results to direct victims to compromised websites.

Though user-initiated activity can be just as dangerous as adversary-initiated activity, it can be more challenging to triage because it often involves unwanted software or riskware, which many organizations deem lower-risk. However, it is critical to respond to this type of activity immediately, as follow-on threats can include infostealers and ransomware.

Top threats relying on user-initiated activity

Several of our top 10 threats—**SocGholish**, **Yellow Cockatoo**, and **Gootkit**—rely on variants of user-initiated activity for initial access. Though not as pervasive, we also saw similar tradecraft with Rose Flamingo, an activity cluster involved in intrusion chains where we later observed various payloads such as STOP ransomware.

- Adversaries behind both Gootkit and Yellow Cockatoo abuse search engine optimization (SEO) to display malicious content at the top of a victim's search results. Because compromised websites are displayed prominently

and presented to the victim from a trusted search engine, victims are often easily "lured" to these sites. They are then prompted to download malicious content masquerading as legitimate content. For example, if a user searched for "this is my query," the binary they downloaded would be named **this-is-my-query.exe**. Because the file looks familiar, users are less likely to scrutinize it closely or look for red flags.

- Rose Flamingo's initial access occurs via file-sharing websites purporting to provide free or "cracked" software.

- Similarly, SocGholish leverages drive-by-downloads masquerading as software updates. SocGholish itself is embedded in legitimate websites that have been compromised to prompt users about the need to download supposed required updates.

In each case, the tradecraft allows the operators to carry out seemingly targeted social engineering intrusions at scale.

TAKE ACTION

To harden your intrusion surface against the search engine tradecraft commonly used by Yellow Cockatoo and Gootkit, we recommend taking steps to prevent access to malicious domains and other malicious content on the internet. This could involve configuring your web proxy to block newly registered and low-reputation domains (e.g., ***.tk**, ***.top**, and ***.gg**) and blocking ads.

To mitigate risks associated with the fake browser updates related to SocGholish and the malicious JavaScript files used by Gootkit, we recommend preventing automatic execution of JavaScript files. You can do this by changing the default file associations for **.js** and **.jse** files.

We also recommend periodically refreshing security training to remind employees of the risks associated with web browsing, as this is discussed less frequently.

TREND

Malicious macOS installers

Malicious installers led to rotten Apples and adware, as macOS systems continued to be targeted in 2021.

We've come a long way from hearing cries of "Macs don't get viruses!," and in 2021, the information security community saw more and more malware targeting macOS systems. In contrast to Windows systems, we observe far fewer malicious documents or email attachments on macOS systems. Instead, the majority of malware we observe on macOS stems from malicious installers that trick victims into thinking they're downloading legitimate content. This approach is particularly insidious, as victims on macOS systems usually possess administrative privileges. **Shlayer**, Bundlore, and **Silver Sparrow** followed this malicious installer trend. Also, four of the eight macOS malware threats Objective-See covered in their **review** of 2021 relied on malicious installers for deployment.

Most macOS threats we observe are malicious adware. Malicious adware is an unwanted program designed to show advertisements on a victim's screen, often within a web browser. A good example of the potential impact of malicious adware comes from the activity cluster Red Canary tracks as Silver Toucan. This cluster discloses its own terms of service that victim hosts may use for proxy activities. Malicious macOS adware often includes tools such as MITMProxy for ad injection, which raises the privacy concern of web traffic inspection on affected hosts.

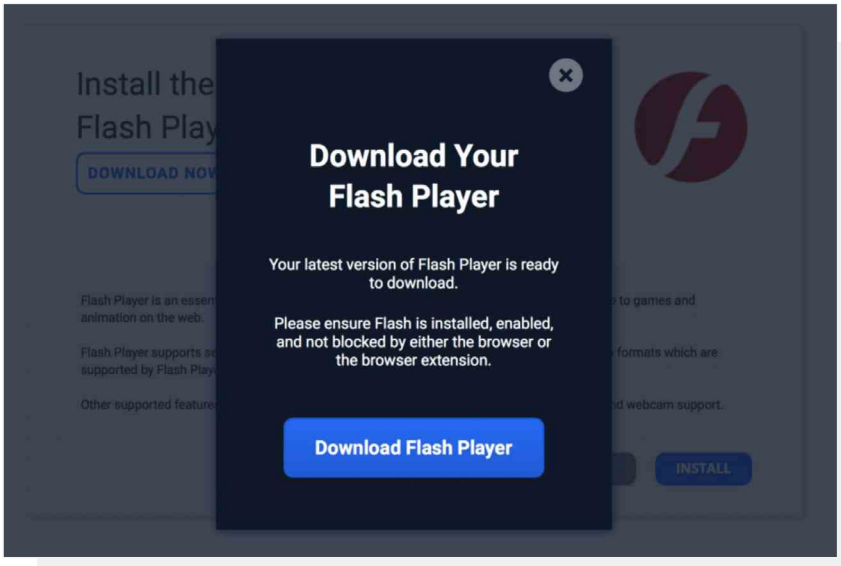


Figure 1: Malicious adware lure

Threats like Shlayer pose as fake Flash Player downloads to look legitimate.

TAKE ACTION

Updating the operating system and applying antimalware controls are the best defenses against malicious software on macOS. Patching to the latest version possible ensures that malware exploits are less likely to succeed. Malware authors still circulate versions of installers that exploit patched vulnerabilities, knowing that not everyone can patch their macOS system. Antimalware controls help mitigate this threat. Where possible, obtain software directly from trusted sources that sign the installers and seek notarization from Apple. Malicious software has been mistakenly notarized in the past, but each case has been rapidly found and remedied.

TREND

Remote monitoring and management abuse

Adversaries continue to use and abuse legitimate remote monitoring and management (RMM) software to move data and control infected hosts.

Adversaries regularly abuse remote monitoring and management (RMM) tools because they're widely used for legitimate reasons and seem benign. Along with the ability to blend in while moving laterally, these tools offer adversaries a reliable way to communicate with and pass information in and out of infected hosts.

In 2021 we identified an uptick of ransomware operators abusing RMM to remotely control victim machines and deploy additional malicious payloads. RMM has typically been used by help desk technicians to resolve issues on client computers. These software suites allow users to remotely control hosts, providing adversaries with a user-friendly graphical interface, secure network connections via cloud hosted infrastructure, and host persistence. This makes it a challenge for defenders to catch the early stages of intrusions. It became increasingly clear to us throughout the year that being able to initially detect abnormal installation and execution of these tools can help thwart ransomware or slow further deployment of malicious payloads.

Not all ransomware operators or affiliates use these tools as part of their intrusion chain, meaning other security controls are still important to cover other access paths. Community reporting has identified ransomware groups like **REvil**, **Conti**, **Avos Locker**, and **Blackheart** using software suites such as ScreenConnect, **Atera**, and Anydesk to gain persistent footholds to hosts after compromising them. In many instances, this led to the deployment of ransomware. Identifying rogue instances of these management tools is a great starting point to help understand and defend your endpoints.

TAKE ACTION

We see the use of RMM tools as a way for adversaries to blend into the vast swath of endpoint telemetry that defenders rely on heavily for finding and eradicating evil. We all need to take a different approach when it comes to detecting this behavior. Rather than solely focusing on blocking known malware samples or writing detection logic surrounding built-in operating system tool abuse (e.g., living off-the-land binaries), keep legitimate third-party software inventory in mind as well.

Enterprises purchase and use hundreds, if not thousands, of software suites, but accounting for what's legitimate in your organization is important. We're not suggesting the near impossible, which is to keep tabs on all abnormal behavior of your numerous applications, but merely suggesting to stay up to date on the permissibility of their presence.

Correlating with the legendary **Pyramid of Pain**, malicious use of RMM tools finds itself near the top of the pyramid, under "Known Tools" and "TTPs." Gathering laundry lists of legitimate software and comparing them against process execution logs will prove valuable for your defensive posture. **SANS** has a great white paper on how defenders can use open source utilities to collect this information remotely from their managed devices. We've also covered this topic more in-depth with multiple detection opportunities in our "**Misbehaving RATs**" blog post.

TREND

Linux coinminers

Coinminers continued to dominate the Linux threat landscape in 2021.

While coinminers affect all operating systems, they made up the majority of the threats we saw on Linux environments in 2021, just as we've seen in years prior. As **Log4j vulnerabilities** consumed the information security news cycle in December 2021, researchers reported adversaries exploiting Log4j to deliver **XMRig** payloads and **other coinminers**. Being able to detect and respond to common threats like coinminers will help any blue team detect a wide range of activity—even when it emanates from unknown exploits.

Many of our Linux coinminer detections began with a Secure Shell (SSH) daemon or a web server process. While we often did not know the exact method of initial access, the intrusion chains we observed suggested that many of them began with weak user authentication or exploitation of web applications. After gaining initial access, adversaries usually leveraged system utilities such as **curl** or **wget** to download additional utilities like shell scripts and coinmining binaries from external sources.

The shell scripts we identified performed various actions, including host reconnaissance, inhibition of competing miners, defense evasion, and persistence. Two common persistence methods we've observed with miner threats like Kinsing and TeamTNT are adding SSH keys to a user's **authorized_keys** file and creating scheduled tasks via the **crontab** command, two relatively easy techniques. A single shell command can be added to a script and establish hooks without much effort on the part of the adversary.

The coinmining binaries that we observed most commonly were XMRig payloads, which were often delivered by adversaries who targeted unpatched endpoints. We observed threats such as **Outlaw** authenticating via SSH to endpoints, presumably as a result of brute-force attempts, followed by executing shell scripts that initiated XMRig payloads named **kswapd0**. We also saw z0miner exploiting **vulnerabilities in Confluence** to deploy XMRig payloads by executing various shell scripts.

Finally, Bird Miner tried to execute XMRig payloads on macOS hosts by **using Qemu to emulate a Linux environment**. No matter how elaborate their initial access techniques, the commonality between these threats is XMRig payloads. Due to its popularity, XMRig artifacts provide excellent opportunities for detection, including several discussed below.

TAKE ACTION

Compromises involving coinmining have been surprisingly consistent over the last few years, and many of the detection opportunities we have **shared previously** are still relevant. Focusing on post-exploitation activity should help, regardless of whether the initial access method is a weak SSH password, outdated web application, or exploitation of a vulnerability like **Log4Shell**.

The best defense against many of the coinminer compromises we observed is patch management. Many of the coinminers we saw exploited flaws in outdated applications like JBoss and WebLogic, so keeping systems updated will deter adversaries who are simply scanning for applications with known vulnerabilities. Strong authentication policies, such as multi-factor authentication (MFA) or locking authentication to just SSH keys, should mitigate techniques like SSH brute forcing.

Here are some additional detection analytics to help identify potential Linux coinminer activity.

Bash **authorized_keys file** modification

This detection analytic will identify instances of Bash processes making file modifications to a user's **authorized_keys** file. Kinsing coinmining malware is one Linux threat that uses this technique for persistence.

process == **bash**

&&

filemod_filepath == **.ssh/authorized_keys**

**Note: There are many shells on Linux endpoints, and this analytic will likely need to be modified to specify the shells that are used within your Linux environment.*

Pkill with **xmr** in command line

This detection analytic will identify processes named **pkill** that have command-line options containing the string **xmr**, which may be observed prior to new XMRig processes executing on infected endpoints.

process == **pkill**

&&

command_line == **xmr**

Renamed coinminers

This detection analytic will identify processes that have command-line options specific to XMRig and similar miners. While command-line arguments can be brittle, this is a great way to catch "lazy" adversaries who do little to hide their activities.

command_line_includes (**stratum** || **--coin** || **--donate-level** || **cryptonight** || **moneropool**)

||

command_line_includes [at least 2 of the following] (**--cpu-priority** || **--max-cpu-storage** || **--algo** || **--url**)

Process connecting to known mining pools

This detection analytic will identify non-web browser processes that initiate network connections to known mining pools.

process != (**chrome** || **firefox** || **msedge** || **iexplore** || **safari**)

&&

network _connection_includes == (**supportxmr** || **xmrpool** || **xmr** || **nanopool** || **monero**)

**Note: This is a non-exhaustive list of pools and web browsers, which you can add to with additional research. Additionally, this analytic will likely need to be tuned to your specific environment, depending on your use of browsers and business purposes.*

TREND

Abusing remote procedure calls

Intrusions leveraging remote procedure calls (RPC) made waves in 2021, particularly PetitPotam and PrintNightmare.

Remote procedure calls (RPC) facilitate local and remote communication between client and server programs. Many Windows services leverage RPCs for communication, and many RPCs expose functions to end users. Depending on privilege levels and the security checks that are (or are not) performed when these functions are implemented, adversaries can abuse RPCs to perform many malicious actions.

We covered RPC abuse in depth **on the Red Canary blog last year**, but two methods of RPC abuse stood out in 2021: PetitPotam and PrintNightmare. Both emerged over the summer, and adversaries quickly adapted them from theoretical proofs of concept for privilege escalation into real-world intrusions. Both were reportedly leveraged in ransomware campaigns, underscoring the urgency behind these threats. We've done extensive testing to replicate these techniques and validate detective and preventive controls for them. What follows is a summary of these compromises and what you can do to defend your organization.

PetitPotam

First published as a proof-of-concept intrusion by researcher **Gilles Lionel** in July 2021, PetitPotam allows an adversary to hijack server authentication sessions and gain access to highly sensitive systems like Active Directory Certificate Services (AD CS). Microsoft published a security bulletin (**CVE-2021-36942**) in August that raised the barrier of entry for PetitPotam, requiring that adversaries first authenticate themselves with legitimate credentials to conduct the intrusion.

PetitPotam enables an adversary to force authentication of a machine by performing an NTLM relay-like intrusion against the **Encrypting File System Remote Protocol (EFSRPC)**, which manages data encrypted by the **Encrypting File System (EFS)** on remote servers. PetitPotam was particularly troubling because the EFSRPC exposed functionality through a DLL (**efslsaext.dll**) that enabled unauthenticated communication through the **LSASS** pipe via the **EfsRpcOpenFileRaw** method. Depending on the patch status, either an unauthenticated or an authenticated adversary can call the EfsRpcOpenFileRaw

method, intercept the authentication response (**NTLM relay**) between the client and a server, and use that to authenticate to another workstation. If they target a domain controller, an adversary could potentially compromise the entire domain by relaying that authentication to an AD CS server. James Forshaw's **detailed article** from August is a great place to learn more.

TAKE ACTION

Security teams seeking to observe and detect PetitPotam intrusions have multiple options. We'll describe relevant telemetry that can be gathered from EDR tools and native operating system logs.

Start by monitoring the **Window Security Event 4624** log for anonymous and other suspicious logins. Many EDR products collect named pipe data, so you can also monitor for **lsarpc** or **efsrpc** named pipe connections to domain controllers. This will show when an unauthenticated user is trying to communicate with the domain controller over those transport protocols.

Microsoft has **published extensive mitigation guidance** describing many controls that administrators can implement to prevent NTLM intrusions in general—some of them more than a decade old—and many of these protections apply to PetitPotam. If it's feasible in your environment, the following can help to mitigate PetitPotam intrusions:

- Update domain controllers and workstations to patch machines against CVE-2021-36942.
- **Disable or set** EFS Service startup type to disabled if service is not being used.
- Enable **SMB signing** to prevent relay intrusions.
- Apply an **RPC filter** to only allow authenticated connection to the EFS service over Kerberos.

PrintNightmare

In July 2021, researchers **Zhiniang Peng** and **Xuefeng Li** disclosed a Windows vulnerability called "PrintNightmare" (**CVE 2021-34527**) that enabled adversaries to perform remote code execution and privilege escalation in two different ways. The objective of each is to connect to a remote host without authentication and cause it to load a malicious DLL. One method abuses the driver installation feature

of the Print System Remote Protocol (MS-RPRN) protocol, while the other abuses a similar driver installation feature of a different protocol, the Print System Asynchronous Remote Protocol (MS-PAR) protocol. In both cases, an inbound connection is accepted by the **print spooler** service (running as SYSTEM), which allows the creation of a separate process also running as SYSTEM. Once an adversary gains SYSTEM level privileges, they effectively have full control over that host.

TAKE ACTION

The following data sources, largely available via commercial EDR tools, can help you identify PrintNightmare-related behavior:

- Monitor files for the the creation of suspicious DLLs in the following file path: **C:\Windows\System32\spool\drivers\(x64/W32X86)*\.dll**
- Monitor module loads to identify when DLLs (especially unsigned ones) are loaded from the following file path: **C:\Windows\System32\spool\drivers\(x64/W32X86)*\.dll**
- Monitor suspicious registry modifications that involve DLLs getting added to the following: **HKLM\System\CurrentControlSet\Control\Print\Environments\Windows x64\Drivers\Version-**.dll** (for x64 systems) or **HKLM\SYSTEM\CurrentControlSet\Control\Print\Environments\Windows NT x86*.dll** (for x86 systems)
- Monitor processes spawning from **spoolsv.exe**. It is unusual for **spoolsv.exe** to spawn child processes under legitimate conditions.

In addition to the above detection opportunities, implement the following controls:

- Update servers and workstations to newest Microsoft releases to patch CVE 2021-34527 and other vulnerabilities.
- **Turn off** the spooler service if it is not being used legitimately.
- Disable Point and Print in the registry: **reg add "HKEY_LOCAL_MACHINE\Software\Policies\Microsoft\Windows NT\Printers\PointAndPrint" /v Restricted /t REG_DWORD /d 0 /f**

TREND

Defense validation and testing

Confirmed testing comprised almost one quarter of our detections in 2021, with many coming from open source tools.

We see a lot of testing. In fact, 23.4 percent of all the confirmed threats we detected in 2021 were confirmed by customers to be testing. We're all for testing (as you can hopefully tell by our work with **Atomic Red Team**), and we wanted to share what we've observed about testing when compared to "proper villains." We also have some suggestions for how to make testing more effective.

In aggregate, confirmed testing behaviors we observed in 2021 differed significantly when compared to non-testing behaviors. When comparing the top 10 detection analytics that appeared in detections marked by customers as testing to those that fired in detections not marked as testing, only three analytics overlapped. Here are some patterns we observed in testing detections during 2021.

Common testing tools

Unsurprisingly, a large volume of the testing detections we observed were from common breach and intrusion simulation tools and open source testing tools. For example, a detection analytic on **CrackMapExec** execution from `cmd.exe` appeared in our top testing techniques, but not in our non-testing detections. CrackMapExec is a post-exploitation tool to audit and assess security in Active Directory environments, so it is a natural choice for testing. This suggests that CrackMapExec is more widely used by testers than non-testers.

Throughout 2021, we also frequently observed **Mimikatz**, **BloodHound**, **Impacket**, **Cobalt Strike**, and **Metasploit** in testing—so much so that testing detections involving these tools helped all of them make it into our **top 10 threats** this year. We consider all of these tools to be "dual-use"—they are used by both adversaries and legitimate users. These dual-use tools present a challenge because it can be difficult to determine if their use is malicious or benign without additional context and understanding of what is normal in an environment. We recommend all organizations have a clear understanding of authorized use of these tools in their environments and treat unconfirmed testing as malicious activity until proven otherwise.

Credential theft methods

We frequently observe credential theft during testing, which is a positive because adversaries frequently do this as well. However, we've noticed that testers often focus narrowly on two approaches for credential dumping. One analytic that fires frequently in testing detections identifies cross-process injection or access activity from **rundll32.exe** to **lsass.exe**. Another analytic identifies instances of **rundll32.exe** dumping process memory using MiniDump, a built-in code library. Part of the reason we observe these behaviors so frequently is because they are integrated into multiple automated breach and intrusion simulation tools, making it more likely for this behavior to occur at scale.

Noisy discovery commands

Another pattern in our testing detections is quick execution of a series of discovery commands such as **ipconfig**, **whoami**, and others. This is in opposition to what we see from many adversaries, who often perform fewer discovery commands in a more targeted way. For example, one of the top analytics we used for detecting testing was for enumeration of Windows Domain Administrator accounts with commands like **net domain admins**. While non-testers use this command as well, we found that testers use it more frequently.

TAKE ACTION

Based on our findings, we encourage organizations to be thoughtful about their testing goals. One approach is to test atomic behaviors without considering the surrounding behavior. This can be helpful to determine if you have the ability to potentially detect that behavior. However, consider also adopting a goal to test a full intrusion chain. This may look different than testing for atomic behaviors—for example, instead of executing 20 discovery commands in quick sequence, you could execute one or two discovery commands followed by other activity, then return to additional discovery commands.

One approach that can help ensure you're testing based on real-world threats that matter is to enhance testing with threat intelligence. Adversary emulation, in which testers use threat intelligence to try to carefully mimic threats of concern as closely as possible, is a widespread methodology that can provide significant value and help organizations improve testing. **MITRE's adversary emulation plans** provide a helpful starting point.

We also recommend changing up your toolset. Automated red teaming and testing tools are powerful, but they are often easier for defenders to detect. To ensure your organization has robust detection capabilities for a range of behaviors, consider different ways you could test the same techniques. For example, instead of just using Mimikatz for credential dumping, try using Gsecdump, NPPSpy, or other tests from **Atomic Red Team**.


2022
THREATS

Top threats

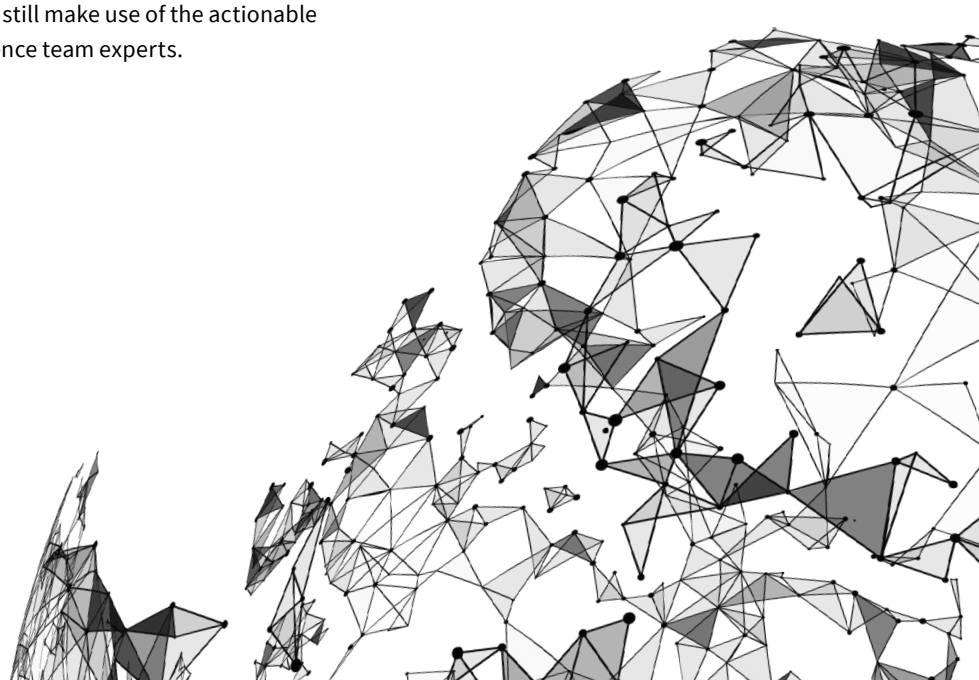
The following chart illustrates the specific threats Red Canary detected most frequently across our customer environments in 2021. We ranked these threats by the percentage of customer organizations affected to prevent a single, major malware outbreak from skewing the metrics.

As discussed in our **Methodology** section, we chose to define "threats" broadly as malware, tools, threat groups, or activity clusters. Eight of our top 10 threats are malware families or tools, while one (TA551) is a threat group named by another team (Proofpoint) and another an activity cluster created by Red Canary (Yellow Cockatoo). This is expected because distinct malware families and tools are often more straightforward to identify, while associating activity to threat groups or activity clusters requires longer-term analysis that may extend beyond the year.

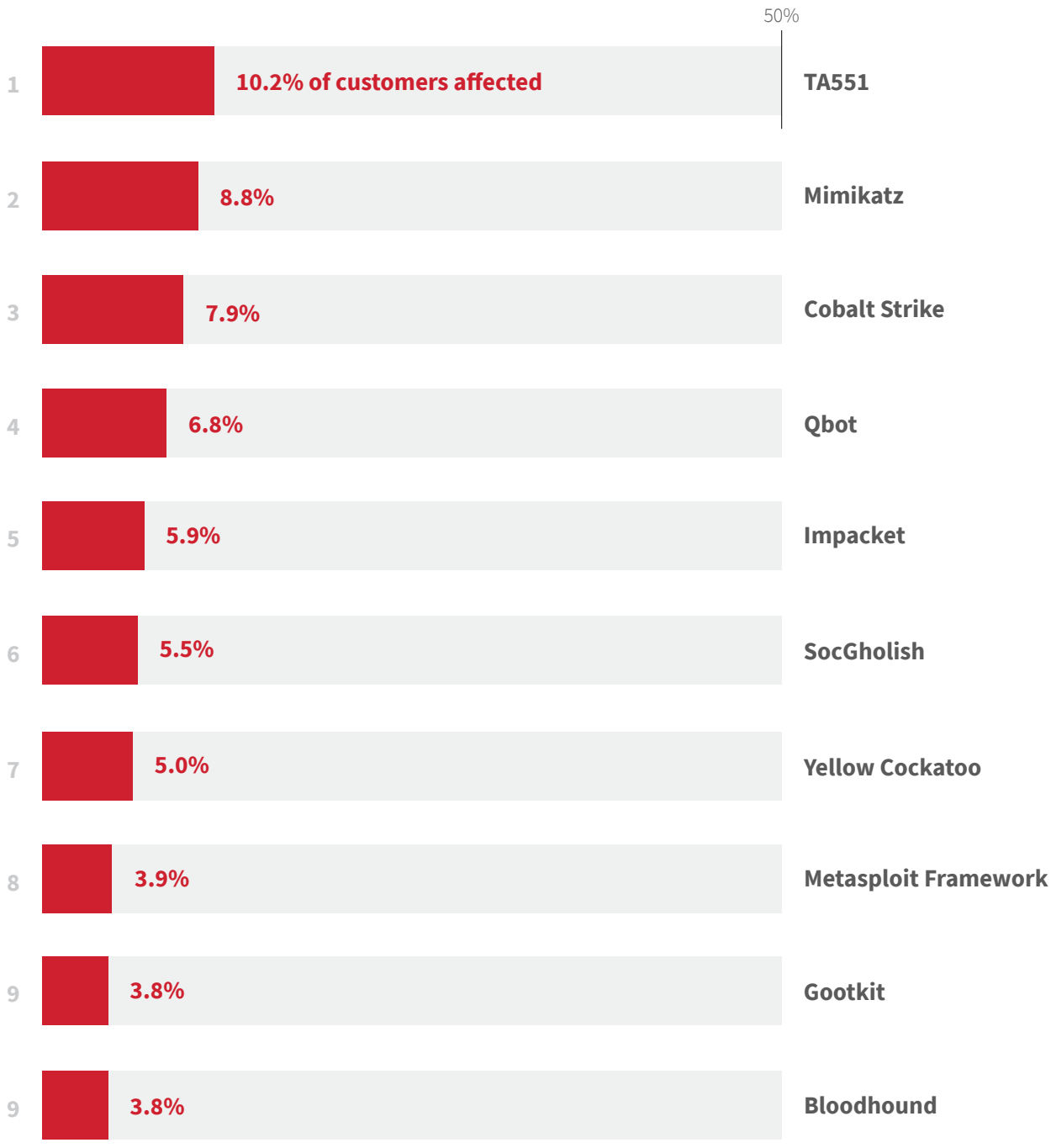
This was our second year tracking top threats. When compared to the top threats in 2020, the overall percentage of customers affected by each threat was down. For example, in 2020, 15.5 percent of customers were affected by TA551, compared to 10.2 percent of customers in 2021. While it's unclear whether this is anything more than a natural ebb and flow of activity, we suspect one factor is the overall increase in detection volume we observed in 2021.

How to use our analysis

These are the most prevalent threats occurring in our customer environments, so we can assume they are prevalent elsewhere. We include advice for responding to each threat and offer detection opportunities so you can better defend your organization. Some defenders may be able to take our detection guidance and apply it directly, while others may not. Regardless, defenders without a detection engineering function can still make use of the actionable analysis of each threat written by our Intelligence team experts.



Top threats



Note: We analyzed each of the top 10 threats in last year's Threat Detection report. However, since there is significant overlap between the top threats for 2021 and 2022, we opted only to analyze new entrants to the top 10 or reanalyze existing top 10 threats that have changed significantly.

TOP TEN THREAT HIGHLIGHTS

Cobalt Strike

Cobalt Strike continues to be a favorite C2 tool among adversaries, as many rely on its functionality to maintain a foothold into victim organizations.

#3

OVERALL RANK

7.9%

CUSTOMERS AFFECTED

Analysis

Cobalt Strike has never been more popular, as adversaries are increasingly adopting it as their favorite C2 tool. Adversaries—ransomware operators in particular—rely substantially on Cobalt Strike's core functionalities as they seek to deepen their foothold in their victims' environments. Its speed, flexibility, and advanced features are likely contributing factors as to why ransomware attacks have been ticking upward in recent years. Some of the most notorious ransomware operators— including groups like **Conti**, **Ryuk**, and **REvil/ Sodinokibi**—are known to rely heavily on Cobalt Strike in their attacks.

The security community is embracing the fact that whatever functional label you place on Cobalt Strike, it's here to stay, it's implicated in all variety of intrusions, and it's our duty to defend against it. Luckily for defenders, over the course of this past year the security community has produced a plethora of great technical analysis and detection opportunities around preventing and investigating Cobalt Strike. Some of the more common detection strategies documented in public reporting include:

- command-line monitoring
- public network infrastructure scanning
- in-memory scanning
- dynamic/static binary analysis
- abnormal process lineage
- network traffic monitoring
- baselining the prevalence of reconnaissance commands

Keep in mind that although many of these methods of detection can be easily bypassed with changes to the Cobalt Strike configurations, we highly suggest using them as a stopgap until your teams develop more advanced methods.

The security community has shared invaluable public resources on analyzing and detecting Cobalt Strike. Our detection opportunities from **last year's Threat Detection Report** remain effective. For defenders getting started with understanding how the tool works and operates, we highly recommend reading

each of the following resources because they all have unique takeaways and cover a majority of the most effective detection techniques:

- https://www.mandiant.com/resources/defining-cobalt-strike-components
- https://blog.talosintelligence.com/2020/09/coverage-strikes-back-cobalt-strike-paper.html
- https://thedfirreport.com/2021/08/29/cobalt-strike-a-defenders-guide/
- https://thedfirreport.com/2022/01/24/cobalt-strike-a-defenders-guide-part-2/
- https://go.recordedfuture.com/hubfs/reports/mtp-2021-0914.pdf

Detection opportunities

Cobalt Strike beacon implant

This detection analytic identifies an adversary using a Cobalt Strike beacon implant to pivot and issue commands over SMB through the use of configurable named pipes. Cobalt Strike beacons have configurable options to allow SMB communication over named pipes, utilizing a host of default names commonly used by adversaries. Analysis should focus on any file modifications to a suspicious named pipe within this process.

file_modifications_include (**pipe\msagent_** || **pipe\interprocess_** || **pipe\lsarpc_** || **pipe\samr_** || **pipe\netlogon_** || **pipe\wkssvc_** || **pipe\srvsvc_** || **pipe\mojo_** || **pipe\postex** || **pipe\status_** || **pipe\msse-**)

Rundll32.exe to spawn SQL Server Client Configuration Utility

This analytic identifies instances of **rundll32.exe** spawning the SQL Server Client Configuration Utility (**cliconfg.exe**). We often see this pattern of process execution when Cobalt Strike leverages DLL Search Order Hijacking as a method of UAC bypass.

parent_process == **rundll32.exe**

&&

process == **cliconfg.exe**

Command-line patterns for Cobalt Strike beacons via GetSystem

This analytic identifies commonly observed command-line patterns when Cobalt Strike beacons escalate privileges via the **GetSystem feature**. Adversaries use **GetSystem** to impersonate a token for the SYSTEM account. This level of access allows an adversary to perform privileged actions beyond that of an administrator.

process == **cmd**

&&

command_line_includes (**/(?i)echo\s+[0-9a-f]{11}\s+\>\;?\s+\\\\\.\\pipe\\[0-9a-f]{6}/.match**)

**Note: The above regular expression will match on the following example what of using GetSystem may look like via a Cobalt Strike beacon:*

C:\Windows\system32\cmd.exe /c echo 92d8cc45954 >; \\.\pipe\446b3c

TOP TEN THREAT HIGHLIGHTS

Impacket

Though Impacket is used legitimately for testing, it is often abused by ransomware operators and other adversaries, thanks in large part to its versatility.

#5

OVERALL RANK

5.8%

CUSTOMERS AFFECTED

Analysis

At its core, Impacket is a collection of Python libraries that plug into applications like vulnerability scanners, allowing them to work with Windows network protocols. These Python classes are used in multiple tools, including post-exploitation and vulnerability-scanning products, to facilitate command execution over Server Message Block (SMB) and Windows Management Instrumentation (WMI). Oftentimes the popular Python scripts **smbexec**, **wmiexec**, or **dcomexec** are used directly without referring to Impacket, as they are versatile and easily implemented code samples. This is the first year that Impacket made it into our top 10 threat rankings, which we attribute to increased use by adversaries and testers alike.

Impacket is an interesting tool as we consider it "dual-use"—it's leveraged by both adversaries and legitimate users. It's often used "behind the scenes" by administration and vulnerability-scanning applications, including Linux tools that manage or scan Windows environments. While Impacket is fairly easy to detect, it can be challenging to determine if it is malicious or benign without additional context and understanding of what is normal in an environment. While threats such as FIN8 malware BADHATCH and multiple ransomware operators have used Impacket, approximately one third of the Impacket detections we saw in 2021 were from confirmed testing. We recommend all organizations have a clear understanding of authorized use of Impacket in their environments, and consider any activity outside of that to be malicious until proven otherwise.

Throughout 2021, operators of Conti, SunCrypt, Yanluowang, Cring, and Vice Society ransomware all used Impacket at some point during intrusions. Impacket acted as a sort of swiss army knife during intrusions, allowing adversaries to:

- retrieve credentials using **secretsdump.py** functionality (SunCrypt)
- issue commands on remote systems during lateral movement (SunCrypt)
- deliver a ransomware binary using **smbexec.py** (Cring and Vice Society)

Responding to Impacket

Response actions may vary depending on the Impacket script component the adversary is leveraging. If you detect a malicious instance of Impacket, seriously consider isolating the endpoint because there's likely an active adversary in your environment.

Once the endpoint is isolated, evaluate if the adversary loaded other tools, if they were able to move laterally from the device, and if they stole credentials. If the adversary moved laterally, isolate any devices they may have accessed. If there is evidence of credential theft, reset passwords for the impacted accounts. Please note that if the adversary leveraged Kerberos, passwords will need a double reset over the course of 10 hours (based on the default 10-hour ticket Time to Live setting) to reset and invalidate existing tickets.

Following the initial response steps above, stop any active processes associated with Impacket, remove any malicious files written to disk, and remove any changes to the device made by the adversary. Reimaging impacted devices is not out of the question, since an adversary may have installed other tools or established persistence.

Detection opportunities

WMIexec execution

This detection analytic uses a regular expression to identify commands from the Impacket **wmiexec** script, which allows a semi-interactive shell used via WMI. This analytic shows output being redirected to the localhost **ADMIN$** share. The regular expression identifies an output file named as a Unix timestamp (similar to **1642629756.323274**) generated through the script.

parent_process == **wmiprvse.exe**

&&

process == **cmd.exe**

&&

command_line_includes (**/(?i)cmd.exe \/Q \/c .*\\\\127.0.0.1\\ ADMIN\$__[0-9]{1,10}\.[0-9]{1,10} 2>&1/**)

SMBexec execution

This detection analytic uses a regular expression to identify commands from the

Impacket **smbexec** script, which allows a semi-interactive shell used through SMB. The regular expression identifies the name of a file share used to store output from the commands for interaction.

parent_process == **services.exe**

&&

process == **cmd.exe**

&&

command_line_includes (**/(?i)cmd.exe \/Q \/c echo cd \^> \\\\127.0.0.1\\[a-zA-Z]{1,}\$__output 2\^>\^&1 > .* & /**)

TOP TEN THREAT HIGHLIGHTS

SocGholish

SocGholish leverages drive-by-downloads masquerading as software updates to trick visitors of compromised websites into executing malware.

#5

OVERALL RANK

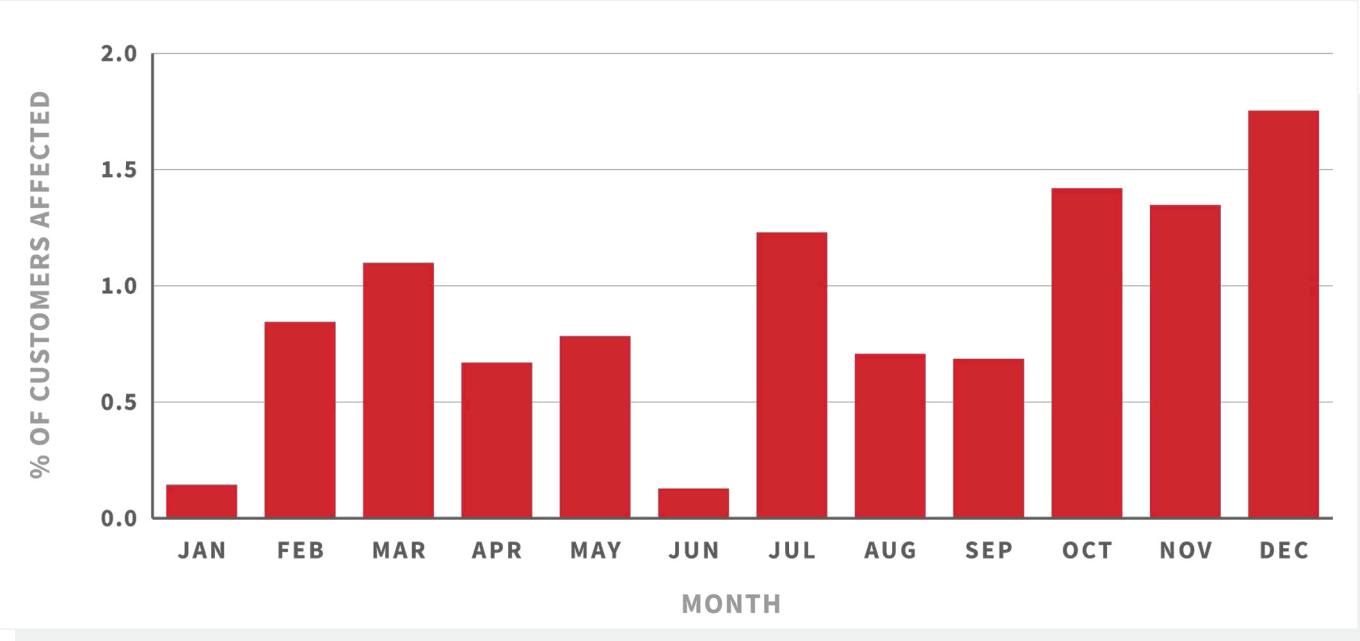
5.5%

CUSTOMERS AFFECTED

Analysis

SocGholish is an initial access threat that leverages drive-by-downloads masquerading as software updates. Active since at least April 2018, SocGholish has been **linked** to the suspected Russian cybercrime group Evil Corp (also known as Indrik Spider). Red Canary encountered SocGholish in a wide variety of industry verticals in 2021. These drive-by-downloads placed SocGholish inside the top five most prevalent threats we track. This ranking was fueled by an increasing number of detections as the year went on, culminating in SocGholish peaking as the most prevalent threat we encountered in December.

Red Canary customers affected by SocGholish in 2021



A SocGholish drive-by-download occurs when an unsuspecting user visits a compromised website and downloads a malicious ZIP file. In one incident described by Expel earlier this year, adversaries compromised an organization's site that was running a vulnerable version of WordPress. Employee endpoints were then infected with drive-by-downloads of SocGholish directly from the company's own website. SocGholish relies on social engineering to gain execution, tricking unsuspecting users into running a malicious JavaScript payload stored within a downloaded ZIP file. These files typically masquerade as browser updates, though other lures include Adobe Flash or Microsoft Teams. Once executed, the JavaScript payload connects back to SocGholish infrastructure, where it shares details about the infected host and can retrieve additional malware.

In 2021, Red Canary observed NetSupport RAT and BLISTER malware delivered by SocGholish. In the past, we have seen SocGholish deploy a Cobalt Strike payload that led to WastedLocker ransomware. The connection between SocGholish and BLISTER is notable, as this malware loader was only identified by Elastic in late December 2021. Following BLISTER deployment in an environment initially compromised with SocGholish, we detected several post-exploitation reconnaissance behaviors on the affected endpoint.

The majority of SocGholish infections we've detected did not result in a second-stage payload, sometimes due to existing mitigations or rapid response to isolate the host. In most cases, we observed reconnaissance activity that only identified the infected endpoint and user. In some cases, Active Directory and domain enumeration followed user discovery. Both of these can be a precursor to lateral movement, however, the hosts were isolated before any lateral movement activity could begin. Much of the reconnaissance conducted by the malicious JavaScript file happens in memory, with data being exfiltrated directly via POST commands to the C2 domain. One good source of insight into this behavior comes from collecting script load content, if such telemetry is available from your endpoint detection and response (EDR) sensor. Collecting this data provides key insight into the specific commands executed and data exfiltrated.

Detection opportunities

JavaScript executing from a ZIP file and making external network connections

Executing script contents from within a ZIP file is unusual, especially when that script is making external network connections. This detection analytic regularly identifies the initial execution and network connections from a SocGholish JavaScript payload extracted from a ZIP file.

process == **wscript.exe**

&&

command_line_includes (**.zip** && **.js**)

&&

has_external_netconn

Script files conducting reconnaissance with **whoami** and writing the output to a file

SocGholish employs several scripted reconnaissance commands. While much of this activity occurs in memory, one that stands out is the execution of whoami with the output redirected to a local temp file with the naming convention **rad<5-hex-chars>.tmp**.

parent_process == **wscript.exe**

&&

process == **cmd.exe**

&&

command_line_includes (**whoami/all>>**)

Enumerating domain trusts activity with **nltest.exe**

Left unchecked, SocGholish may lead to domain discovery. This type of behavior is often a precursor to ransomware activity, and should be quickly quelled to prevent further progression of the threat.

process == **nltest.exe**

&&

command_line_includes (**/domain_trusts** || **/all_trusts**)

TOP TEN THREAT HIGHLIGHTS

Yellow Cockatoo

Yellow Cockatoo is an activity cluster involving a remote access trojan (RAT) that filelessly delivers various other malware modules.

#7

OVERALL RANK

4.9%

CUSTOMERS AFFECTED

Analysis

As Yellow Cockatoo uses effective search engine poisoning tactics, can stealthily persist in a compromised environment, and appears to affect a wide array of organizations across various sectors and geographies, we weren't surprised to see it crack our top 10 threats in 2021. In September the volume of Yellow Cockatoo detections increased substantially (relative to earlier in the year). This may have been the result of a new installation mechanism, chronicled in detail by researchers from Morphisec (they call this threat "Jupyter").

While much of the public reporting, notably a robust profile published by Morphisec, covers an infostealer component of Yellow Cockatoo, we often observe behaviors that occur earlier in the Yellow Cockatoo kill chains. This typically includes installation mechanisms, which deliver code that runs persistently. This code later downloads and executes additional modules that are never written to disk. In many of the instances of Yellow Cockatoo activity we observed, the payloads were a minimal version of the original components documented by Morphisec, with the infostealer functionality delegated to additional modules.

Yellow Cockatoo tradecraft is wide-ranging, and there are several variations on its attack chain. Over time, the most significant detection opportunities stem from the behaviors we observe consistently. These include but are not limited to the tradecraft outlined below.

Initial access: Search engine redirects enable Yellow Cockatoo operators to perform seemingly targeted social engineering attacks at scale. Initial access by Yellow Cockatoo often occurs via a search engine redirect that directs a user from a legitimate search engine to a site that downloads a malicious file bearing the victim's search query as its name (for example: "this-is-my-search-query.msi" or "this-is-my-search-query.exe"). Because potential victims are directed to a site based on a search they initiated, they may be more inclined to engage with its content. Though many adversaries craft tailored attacks and leverage familiar themes, Yellow Cockatoo is unique in its ability to dynamically "customize" its attacks based on victims' real-time searches.

Execution: Following installation, the EXE or MSI file spawns a command line and creates a similarly named TMP file that launches PowerShell. All of this is precursor activity that leads to the execution of a malicious dynamic link library (DLL). This is a remote access trojan (RAT) implemented as a .NET assembly designed to be reflectively loaded into PowerShell.

Defense evasion: Since Yellow Cockatoo's follow-on activity occurs in memory, it poses a unique challenge to defenders. Yellow Cockatoo uses XOR and Base64 encoding to ensure its files are obfuscated and do not exist in cleartext on disk. Cleartext is only present in memory and only exists after it is invoked by its loader. Accordingly, static detection rules for files on disk may miss malware components.

To harden your attack surface against the search engine redirects commonly used by Yellow Cockatoo, we recommend taking steps to prevent access to malicious domains and other malicious content on the internet. This could involve configuring your web proxy to block newly registered and low-reputation domains (e.g.,*.tk, *.top, and *.gg) and block advertisements.

Detection opportunities

PowerShell writing startup shortcuts

We frequently observe adversaries using PowerShell to write malicious LNK files into the startup directory to establish persistence. Accordingly, this detection opportunity is likely to identify persistence mechanisms in multiple threats. In the context of Yellow Cockatoo, this persistence mechanism eventually launches the command-line script that leads to the installation of a malicious DLL:

process == **powershell.exe**

&&

command_line_includes (**appdata**)

&&

filemod_path_includes (**start menu\programs\startup**)

&&

filemod_extension == **.lnk**

**Note: You can test the efficacy of this detection opportunity by running this Atomic Red Team test in PowerShell with elevated privileges.*

PowerShell utilizing System.Reflection.Assembly to load a DLL

This detection analytic identifies PowerShell using **System.Reflection.Assembly.** to load a DLL. However, this analytic may generate false positives in your environment and likely requires tuning.

```
process == powershell.exe

&&

command_line_includes ( reflection.assembly )

&&

command_line_regex_encoded == /(?i):\(?load\)?(?:|file)\(/
```

TOP TEN THREAT HIGHLIGHTS

Gootkit

Gootkit is a banking trojan that can deliver additional payloads, siphon data from victims, and stealthily persist in a compromised environment.

#9

OVERALL RANK

3.8%

CUSTOMERS AFFECTED

Analysis

A malware threat with a JavaScript loader component, Gootkit has been actively observed in the wild for more than a decade. Over the past several years, it has evolved into a multi-stage tool used to facilitate a range of hands-on-keyboard activity in multi-pronged attacks, wherein more than one objective is likely accomplished. Gootkit was **originally** delivered via spam email campaigns and older exploit kits, but over time its initial access has shifted to SEO poisoning tactics. Specifically, operators alter search engine results to direct victims to **legitimate but compromised websites** hosting Gootkit. Upon visiting a compromised website, victims are prompted to download a ZIP archive containing a malicious JavaScript file, which if executed can allow an adversary to remotely access a victim's system. While some researchers **track** the delivery mechanism as "Gootloader" and the trojan activity as "Gootkit," Red Canary tracks both components as "Gootkit." Our classification may shift as we gather additional information.

Follow-on activity varies. In 2021, Red Canary saw operators use Gootkit to deliver Cobalt Strike. Though we didn't observe any ransomware in that intrusion, the intrusion chain mirrored public reporting of compromises where victims' networks were ultimately encrypted with Sodinokibi (REvil) ransomware. Based on **public research** and follow-on activity observed in customer environments last year, it's likely that Gootkit operators facilitate ransomware-as-a-service (RaaS) activity in some cases, either deploying other payloads directly or selling access to environments with Gootkit infections. We have also observed Gootkit dropping the Osiris banking trojan.

While we've observed Gootkit detections in customer environments across multiple sectors, almost without exception, infections occurred after victims visited compromised websites purporting to host content related to legal or financial agreements. Victims were **likely** directed to these sites after initiating queries in common search engines with keywords such as "agreement," "contract," and the names of various financial institutions. Given the volume of Gootkit detections and the range of victims, this threat is likely more opportunistic than targeted to a specific industry or organization. Accordingly, Gootkit remains a threat to all organizations.

One hypothesis as to why we observe Gootkit so frequently is that it is downloaded from sites victims navigated to based on search results they initiated themselves, as we further discuss in the user-initiated initial access section.

Detection opportunities

Windows Scripting Host executing JavaScript files

This detection analytic will identify unusual activity originating from **wscript.exe** executing JavaScript files from the **%APPDATA%** directory. This applies to GootKit because the initial loader for the threat is implemented in JavaScript that gets executed via **wscript.exe** when the victim double-clicks on the downloaded loader.

process == **wscript.exe**

&&

file_path_includes (**%APPDATA%**)

PowerShell using a shortened **EncodedCommand** flag

This detection analytic will identify use of the shortened **EncodedCommand** flag in PowerShell, a tactic often used by Gootkit operators and others to obfuscate malicious code on an endpoint. Like all detection analytics, this may generate some false positives in your environment that require tuning. This applies to GootKit at multiple stages after the loader, when this threat uses PowerShell to deobfuscate and execute downloaded payloads.

process == **powershell.exe**

&&

command_line_includes == [any variation of the **-encodedcommand** switch]*

Note: the encoded command switch has many variations, including* **-encodedcommand, **-e**, **-enc**, *and many other variations*

TOP TEN THREAT HIGHLIGHTS

BloodHound

BloodHound is an open source tool that provides visibility into Active Directory environments. It is a common precursor to follow-on activity, whether that's further testing or ransomware.

#9

OVERALL RANK

3.8%

CUSTOMERS AFFECTED

Analysis

BloodHound is an open source tool that can be used to identify attack paths and relationships in an Active Directory (AD) environment. Like **Impacket**, this is the first year BloodHound made it into our top 10 threat rankings, thanks to both testing activity and adversary use. It is popular among adversaries and testers because having information about an AD environment can enable further lateral movement throughout a network. BloodHound has multiple components, including SharpHound, which is a data collector for BloodHound written in C#. Throughout 2021, SharpHound was one of the most common BloodHound components we observed.

Multiple adversaries used BloodHound during 2021, including **FIN12** and operators of Yanluowang ransomware. We also observed BloodHound being used by operators in conjunction with **Cobalt Strike** only 75 minutes after a user first opened a malicious XLS phishing lure that initiated a **SquirrelWaffle** malware payload.

Because adversaries often leverage BloodHound early in their intrusion, defenders should be prepared with robust detection and a quick response to stop the malware in its tracks. BloodHound's role as a dual-use tool can make it particularly challenging to determine if its presence is authorized or malicious, meaning that a solid understanding of its allowed use in an environment is critical to respond appropriately.

Identifying SharpHound components gathering data can be challenging. To gather AD data, SharpHound connects to multiple hosts over ports 137 and 445, along with multiple named pipe connections. As your environment scales larger, the noise from SharpHound will scale accordingly. For most organizations, SharpHound activity will likely appear to be SMB scanning activity until investigated further.

Detection opportunities

High-volume port 445 connections

This detection opportunity identifies a single process exceeding a set threshold for a normal volume of network connections to port 445. We did not specify logic for this detection analytic, since the normal number of connections will differ in each environment. While it takes some tuning, this analytic helps detect not only BloodHound, but also various types of post-exploitation SMB scanning and lateral movement.

Common BloodHound command-line options

This detection analytic identifies processes that contain common command lines consistent with the execution of BloodHound. While this is a simple analytic, we've found it to be effective in identifying BloodHound. It's a good supplement to the port 445 analytic, which can require more tuning.

> command_line_includes (**-collectionMethod** || **invoke-bloodhound** || **get-bloodHounddata**)

THREAT: NEW ACTIVITY CLUSTER

Rose Flamingo

Rose Flamingo relies on search engine optimization (SEO) poisoning to trick victims into infecting themselves.

#29

OVERALL RANK

1.1%

CUSTOMERS AFFECTED

Analysis

Rose Flamingo is an activity cluster named by Red Canary that focuses on opportunistic, financially motivated malware as an initial access broker. Rose Flamingo targets victims who are looking to download licensed software without having to pay for it. Payloads related to Rose Flamingo typically arrive as archive files that are distributed via phony file-sharing websites purporting to provide users with "free" cracked software packages. To lure potential victims, the adversaries behind Rose Flamingo use **search engine optimization (SEO) poisoning** to elevate a malicious site's search ranking.

Rose Flamingo victims will typically download a ZIP archive file containing two files at a minimum. Archives related to Rose Flamingo may contain words like `free`, `key`, `download`, `license`, `latest`, `ISO`, and `crack`. While these archives usually appear as ZIP files, they infrequently appear as other compressed archive formats as well. The files in a typical Rose Flamingo archive are a "password" text file and one password-protected archive. Some iterations of these "password" files contain the password and some classic ASCII art, as shown below, though the purpose behind the art is unknown. This type of delivery method conceals the malicious payloads that are contained within the password-protected archive from any prying security software

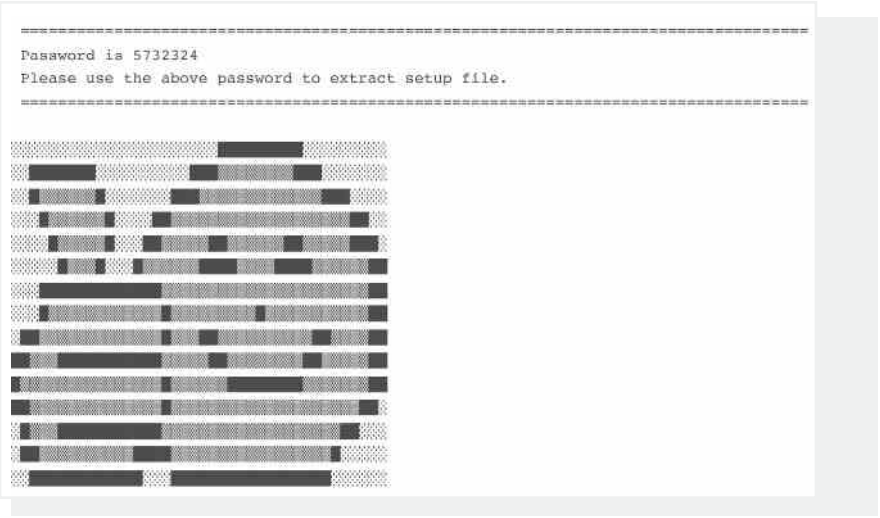


Figure 2: The contents of a "password" text file associated with Rose Flamingo

While we created the Rose Flamingo naming convention to help us track activity we consider to be related, there's a growing body of external research documenting components that partially overlap with what we define as Rose Flamingo. Much of this emerging research dropped in 2021, and it's worth reviewing for anyone who is interested in learning more about related activity:

- In January 2021, CSIS Security Group **released research** referencing infrastructure and payloads that overlap with Rose Flamingo.
- In March 2021, Fortinet detailed a threat called **Netbounce**, which uses a similar file-naming convention and has some overlapping infrastructure.
- Just about a week later in March 2021, Proofpoint published research about a threat they call **CopperStealer (Mingloa)**, describing infrastructure and payload-naming conventions that are very similar to Rose Flamingo's.
- In June 2021, an Ahnlab report described a threat called **Cryptbot**, detailing files used for delivery that appear to overlap with Rose Flamingo's file-naming conventions.
- In late July 2021, BitDefender joined the party, helping corroborate many of our own observations with their **MosaicLoader whitepaper**, a great report that touches on much of the initial loader activity we've observed in Rose Flamingo-related incidents.
- Last but not least, in September 2021, SophosLabs released research that focuses on a **content delivery network** that has many infrastructure and payload overlaps with our analysis.

Because none of us have perfect visibility, we appreciate that other teams share their perspective and how they track these threats.

As seen in our Intelligence Insights rankings from month to month, Rose Flamingo made our top 10 list for the first time in July 2021, climbing to eighth place for most prevalent threats that month. It also made our top 10 in **September** and **October 2021**. Red Canary has observed Rose Flamingo delivering various stealers such as Cryptbot, RedLine, and Raccoon, in addition to more concerning payloads such as **STOP ransomware**. We will continue birdwatching as we look for new opportunities to observe and take action when threats like Rose Flamingo find a new place to roost.

Detection opportunities

Archive containing ZIP and TXT files containing `password`

This detection analytic will identify processes making file modifications for ZIP archive files and TXT files with the string `password` in them, which we commonly observe in Rose Flamingo activity. The password files may contain different naming variations, such as `p@ssword` or `passw0rd`. Detecting TXT files with these strings may generate fewer false positives. If you have trouble getting this detection opportunity to work, you may find further success focusing on **application** processes that are responsible for handling archives in your organization, such as **7zip**.

filemod_includes (`zip`)

&&

filemod_includes (`password` && `txt`)

Potential Rose Flamingo loader

This detection analytic will identify unusual processes that contain naming schemas that have been observed in use by loaders related to Rose Flamingo archives.

process_name_includes (`main-` || `installer-v` || `main_setup_x86x64` || `x86_x64_setup` || `setup_x84_x64`)

Potential Rose Flamingo loader

This detection analytic will identify files and file paths that contain strings commonly observed in archives delivered by Rose Flamingo.

filepath_includes (`-free` || `-crack` || `-download` || `-key` || `-license` || `-iso` || `-Install`)

||

filename_includes (`-free` || `-crack` || `-download` || `-key` || `-license` || `-iso` || `-Install`)

&&

filename_includes (`zip` || `7z` || `rar`)

THREAT: NEW ACTIVITY CLUSTER

Silver Sparrow

Silver Sparrow is a macOS activity cluster with fully functional distribution methods and infrastructure but no final payload.

In February 2021, Red Canary discovered an activity cluster we named **Silver Sparrow** when we identified a strain of macOS malware using a `LaunchAgent` to establish persistence. Distributed via downloads from AWS S3 buckets, malware dropped by Silver Sparrow relies on installation through macOS PKG files. We analyzed two versions of Silver Sparrow malware: The first version contained a Mach-O binary compiled for Intel x86_64 architecture only, and the second version included a Mach-O binary compiled for Intel x86_64 and M1 ARM64 architectures. The downloader was novel because of the way it used JavaScript for execution and the appearance of a related binary compiled for Apple's new M1 ARM64 architecture. During installation, the malware executed JavaScript commands to orchestrate the creation of files and scripts for persistent execution. These files attempted to download a future payload determined by a file from an additional S3 bucket retrieved every hour.

Since we observed multiple files and components on victim machines, we decided to cluster all the suspicious artifacts under the Silver Sparrow activity cluster, including an unusual `._insu` file that seems to instruct the malware to remove itself from an endpoint.

Thanks to our friends at **MalwareBytes**, we determined that the Silver Sparrow activity cluster affected tens of thousands of macOS endpoints across 164 countries as of February 2021, including high volumes of detection in the United States, the United Kingdom, Canada, France, and Germany. Although we never observed Silver Sparrow delivering additional malicious payloads, it was operationally mature and affected many thousands of machines worldwide.

Overall, Silver Sparrow is interesting and unique because:

- At the time of analysis, its malware was compatible with M1 ARM64 and Intel chipsets. Researchers have uncovered very few threats for the M1 ARM64 architecture because the architecture is young.
- Its installer packages leverage the macOS Installer JavaScript API to execute suspicious commands. This is the first malware we've seen do this.
- Its infrastructure was hosted on AWS S3, making it hard to block outright. The decision to use AWS infrastructure suggests an operationally mature adversary.

Detection opportunities

PlistBuddy utility manipulating **LaunchAgent**

The **PlistBuddy** command is a built-in tool in macOS that allows administrators to manipulate property list, or plist, files used to configure various parts of the macOS operating system. Silver Sparrow used the command to manipulate **LaunchAgent** plists and allow persistence. **PlistBuddy** with the command line including **RunAtLoad** indicates an adversary is specifically manipulating a **LaunchAgent** or **LaunchDaemon**'s capability to execute code at boot.

process == **PlistBuddy**

&&

command_line_includes (**RunAtLoad**)

Sqlite3 loading the Quarantine file

The Quarantine feature of macOS prevents certain file types from easily executing after being downloaded from the internet. The system keeps a record of all downloaded files in a SQLITE database at **~/Library/Preferences/com.apple.LaunchServices.QuarantineEventsV***. Silver Sparrow malware and other macOS threats commonly query this record using the **sqlite3** command to determine where they were downloaded from to report back to the adversary for metrics (i.e., whether or not the deployment path was successful).

process_name == (**sqlite3**)

&&

command_line_includes (**LSQuarantineURLString**)

TAKE ACTION

We included some detection opportunities to help identify Silver Sparrow activity. Also, see the macOS trends page for defense strategies to protect yourself from macOS threats.

RELEVANT THREATS OF 2021

Bazar

The Bazar family of malware continued to be active in 2021, spurring ransomware infections.

#16

OVERALL RANK

1.7%

CUSTOMERS AFFECTED

The Bazar malware family was quite active in 2021, spreading via multiple delivery affiliates, including **TA551** and BazaCall. There are many names for Bazar (sometimes referred to as "Baza") floating around that refer to various parts of the intrusion chain. Bazar is relevant because of its role as a malware precursor, and many 2021 intrusions starting with Bazar led to ransomware like **Ryuk** and **Conti**. The Bazar malware family encompasses a loader, BazarLoader, and backdoor, BazarBackdoor. These components have been delivered via multiple delivery affiliates. As we discuss in the Affiliates section of this report, differentiating initial delivery affiliates from loaders and payloads will help you understand each phase of the threat and how to better protect your organization.

One affiliate we've been tracking for a while, TA551, began delivering Bazar during 2021. While TA551 relied on email attachments to deliver Bazar, another affiliate behind a 2021 phishing campaign known as **BazaCall** opted to trick users into calling a phone number sent in an email. After a victim called the number, an adversary provided step-by-step instructions that led to downloading Bazar malware. (Check out **Brad Duncan's video** for an example of how this intrusion plays out.) Once BazaLoader was installed, BazaCall led to Cobalt Strike and eventually, ransomware.

Detection opportunities

Microsoft Certificate Services using **certutil.exe** to initiate download

This detection analytic looks for instances of the Microsoft Certificate Utility (**certutil.exe**) initiating a download, a technique used to download Bazar payloads.

process == **certutil.exe**

&&

command_line_includes (**urlcache**)

RELEVANT THREATS OF 2021

Latent threats

Threats come and go, but some—like USB stowaways and network worms—like to stick around.

Latent threats demonstrate that most adversaries do not need to be advanced or sophisticated to execute code or persist in an organization. They can simply settle to be an **adequate persistent threat**, using techniques and artifacts that virtually anyone can find. This section outlines opportunities to detect and respond to tried-and-true threats like USB stowaways and network worms.

USB stowaways

In this section we characterize "USB stowaways" as threats that leverage USB thumb drives to find their victims.

Floxif (ranked #75 in 2021)

Floxif, short for "FloodFix," is a type of file-infecting malware that researchers have observed spreading to **some of the farthest reaches** of organizations' networks **since 2012**. In the detections that we've observed, Floxif most commonly arrived on endpoints via USB thumb drive. Floxif self-replicates by identifying processes running in memory that are eligible for infection and replaces them with new, Floxif-compromised binaries. Many variants of Floxif malware rely on writing the accompanying DLL `symsrv.dll` to a unique location, so detecting this threat can be done with relatively high confidence

Floxif DLL file modification

This detection analytic identifies file modifications that are consistent with Floxif malware execution.

file_modification_includes (`Common files / System / symsrv.dll`)

Gamarue (ranked #12 in 2021)

Gamarue is a malware family that was first observed by researchers in 2011 and rendered inactive after a **joint takedown operation** in **2017**. While many variants of **Gamarue** exist, the variant we observed most frequently in 2021 was an Autorun worm that spread primarily via infected USB drives. This is no different from what we saw in 2020, and we expect to continue seeing it for as long as users keep deploying infected USB drives to ferry files from one endpoint to another.

Explorer launching Rundll32 without any DLL in the command line

```
process == rundll2.exe
&&
parent_process == explorer.exe
&&
command_line_does_not_include (.dll)
```

Conficker (ranked #28 in 2021)

Bridging the gap between USB worms and network worms, Conficker is a worm that feverishly spread across the internet in late 2008, leveraging the NetBIOS vulnerability **MS08-067**. As more sophisticated variants were developed, USB Autorun worming functionality was soon baked into Conficker as well, helping to further spread this worm via **sneakernet**. Fourteen years later, Red Canary still detects artifacts related to Conficker, most of which are leftover persistence mechanisms from incomplete remediation. While antivirus products may be doing most of the heavy lifting in terms of remediating active instances of Conficker, those errant scheduled tasks may still be out there trying to launch Conficker DLLs of bygone days.

Rundll32 executing with command lines consistent with Conficker

This detection analytic will identify unusual activity originating from the `rundll32.exe` process. Werfault typically spawns with command-line parameters when a process crashes, providing the program with input to create an error report. If you are having trouble getting this detection opportunity to work in your environment, you may find additional success by focusing only on

processes where **taskeng.exe** or **svchost.exe** are the parents of Rundll32.

```
process _name == rundll32.exe
&&
command_line == rundll32\.exe [a-z]{5,8}\.[a-z]{1,3},[a-z]{5,8}
```

Network worms

In this section we characterize "network worms" as threats that exploit vulnerabilities in software to infect and establish control over an endpoint. Following initial access, adversaries leverage the infected endpoints' network connections to identify additional assets to infect and repeat the cycle.

WannaCry ransomware (ranked #31 in 2021)

WannaCry, often shortened to "WCry," is a ransomware variant that spreads as an SMB worm leveraging the **ETERNALBLUE** vulnerability, **MS17-010**. WannaCry was first observed in May 2017, indiscriminately spreading across many organizations. Early variations of WannaCry had code built in to discontinue ransomware operations, but later versions of did not include this functionality. Half a decade later, some might laugh that we're including WannaCry in a report released now, and we must admit that seeing WannaCry so high in our rankings was a bit of a shock for us too, but here we are. Simply put, there's a reason why the vulnerability WannaCry targets, **MS17-010**, is known as ETERNALBLUE. Just like **MS08-067** is to Conficker, **MS17-010** is so reliable that we are likely to continue seeing WannaCry for quite some time. If you are concerned that your endpoints might be afflicted by WannaCry, Microsoft provides guidance on how to identify endpoints that may be **susceptible** to SMBv1 exploitation, as well as **mitigation techniques** that are still **applicable** today.

Process names that are consistent with WannaCry binaries

This detection analytic will identify processes that are executing with process names that are consistently observed in use by WannaCry binaries.

```
process_name == mssecsvc.exe
||
process_name == tasksche.exe
```

LSASS spawning processes

This detection analytic will identify instances of the **Local Security Authority Subsystem Service** (**lsass.exe**) spawning processes that are not typically observed being launched by **lsass.exe** . LSASS is an injection target for WannaCry, as detailed by **Microsoft**.

parent_process == **lsass.exe**

&&

process_name != **werfault.exe** || **lsass.exe**

WannaMine cryptominer (ranked #57 in 2021)

WannaMine, a portmanteau of WannaCry and Mine, is a malware family that focuses on deploying coinmining payloads. The “Wanna” part of the name of this threat comes from the use of the same ETERNALBLUE vulnerability that WannaCry leveraged. While WannaMine may be **old news** to some, Red Canary observed new infections throughout the course of 2021. There’s a reason why vendors are still producing articles providing guidance around WannaMine **cleanup and remediation**.

PowerShell executing with NoProfile and NonInteractive CLI parameters

This detection analytic identifies instances of **powershell.exe** executing with the strings **-nop** and **-noni** in the command line, which are shortenings for the PowerShell **parameters** **NoProfile** and **NonInteractive** . These command-line parameters are rarely observed together legitimately, making for another analytic that can be used to identify a **multitude** of threats, not just **WannaMine**.

process == **powershell.exe**

&&

command_line_includes (**-nop** && **-noni**)

Honorable mention

Zloader is neither a "USB stowaway" nor a "network worm," and though it never causes enough of a stir to breach our top rankings, it still deserves an **honorable mention** as a latent threat. The adversaries behind Zloader typically devise **innovative ways** to reach their victims before making the news with their next campaign, yet even with the latest passing headline, they often still rely on **less novel TTPs** that can give their presence away.

Zloader (ranked #53 in 2021)

Zloader is a banking trojan that has targeted victims through a variety of avenues since 2016. Though its TTPs have changed over the years, the driving force behind Zloader continues to appear to be financial motivation. In mid 2020, Zloader's operators began delving into delivering **ransomware** alongside their more typical banking trojan payloads, elevating our concern whenever we detect a threat that is consistent with Zloader activity. Zloader makes this list because it is another threat that you may not hear much from for a few months but is always likely to creep its way back.

PowerShell modifying Windows Defender exclusions

This detection analytic identifies instances of PowerShell issuing commands to modify Windows Defender exclusion policies. This activity is consistent with ZLoader activity that occurs prior to the execution of follow-on payloads. Additional threats, such as Purple Fox, leverage this TTP as well.

```
process == powershell.exe
&&
command_line_includes (Add-MpPreference || Set-MpPreference)
&&
command_line_includes (ExclusionExtension || ExclusionPath ||
ExclusionProcess)
```

TAKE ACTION

Even if you're not seeing them in headlines, it is important to evaluate threats that have been known to be problems in the past. If your least favorite adversary has gone dormant, there's a chance that they may come back using many of the same TTPs. Make sure your endpoints are up to date with the latest patches, and if you find yourself to be afflicted by the many threats that we have outlined today that abuse Autorun functionality, you may want to consider **disabling Autorun** across the organization.

2022

TECHNIQUES

Top techniques

The purpose of this section is to help you detect malicious activity in its early stages so you don't have to deal with the consequences of a serious security incident.

The following chart represents the most prevalent **MITRE ATT&CK®** techniques observed in confirmed threats across the Red Canary customer base in 2021. To briefly summarize what's **explained in detail in the Methodology section**, we have a library of roughly 3,000 detection analytics that we use to surface potentially malicious and suspicious activity across our customers' environments. These are mapped to corresponding MITRE ATT&CK techniques whenever possible, allowing us to associate the behaviors that comprise a confirmed threat detection with the industry standard for classifying adversary activity.

TOP TECHNIQUES

NAME	TECHNIQUE RANK (SUB-TECHNIQUE RANK)	% OF CUSTOMERS AFFECTED
T1059: Command and Scripting Interpreter • T1059.001: PowerShell • T1059.003: Windows Command Shell	**1** (1) (2)	**53.4%** (35.0%) (28.1%)
T1218: Signed Binary Proxy Execution • T1218.011: Rundll32	**2** (3)	**34.8%** (23.3%)
T1047: Windows Management Instrumentation	**3**	**15.4%**
T1003: Credential Dumping • T1003.001: LSASS Memory	**4** (6)	**18.3%** (13.3%)
T1105: Ingress Tool Transfer	**5**	**20.4%**
T1055: Process Injection	**6**	**21.7%**
T1053: Scheduled Task/Job • T1053.005: Scheduled Task	**7** (4)	**14.7%** (12.2%)
T1027: Obfuscated Files or Information	**8**	**19.4%**
T1036: Masquerading • T1036.003: Rename System Utilities • T1036.005: Match Legitimate Name or Location	**9** (7) (11)	**22.1%** (15.6%) (7.9%)
T1574: Hijack Execution Flow • T1574.001: DLL Search Order Hijacking	**10** (5)	**8.4%** (7.8%)

Note: We chose not to include analysis for each technique in the PDF supplement to the report, but, as always, they're available in full on the* **Threat Detection Report website.

What's included in this section?

We've written extensive analysis of 12 ATT&CK techniques and sub-techniques. Each technique-specific section includes:

- a brief **analysis** of how and why adversaries leverage a given technique
- descriptions of data sources that offer **visibility** into the technique (e.g., command monitoring, process monitoring, etc.)
- guidance on the tooling or logs that will enable you to **collect** those data sources (e.g., EDR, Sysmon, AMSI, Windows Events. etc.)
- specific examples of how you can use that telemetry to **detect** adversaries abusing the technique
- individual **tests** for emulating how adversaries abuse the technique to validate that you can observe or detect it

The bottom line

Examined holistically, the list of prevalent techniques showcased in this report suggests that if you can detect threats relatively early in the intrusion lifecycle, you're much less likely to face the consequences of a significant cyber attack. This principle has saved many of our customers from immeasurable grief over the years.

To that point, we mostly detect adversaries as they're setting the stage for later, more impactful actions. We catch them attempting to abuse native operating system utilities to execute code or bring in custom tooling. We catch them elevating their privilege levels to get deeper access to compromised systems. We catch them establishing persistence so they can maintain their presence. We catch them manipulating our customers' defensive controls to evade prevention or detection. These are necessary means to an end—whether the goal is to conduct espionage, a ransomware attack, or something else altogether. When we disrupt these means, we prevent their ends.

This is precisely why exfiltration and impact techniques (e.g., ransomware) don't rank highly on our list. The following heatmap shows the distribution of the 20 most prevalent techniques across the ATT&CK matrix.

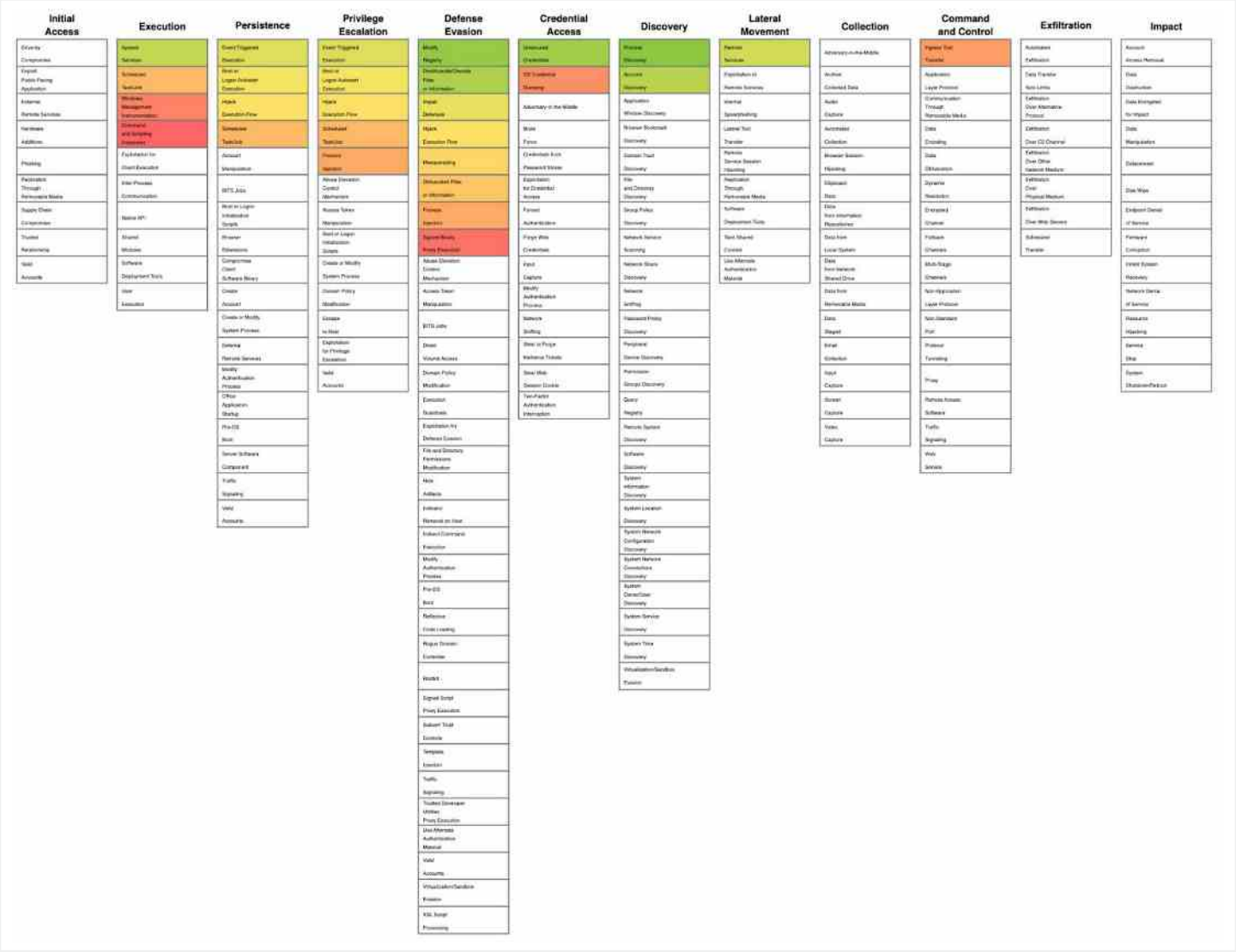


Figure 3: MITRE ATT&CK Navigator layer showing the 20 most prevalent ATT&CK techniques detected by Red Canary

This isn't to suggest that we never encounter ransomware. In fact, we routinely encounter **ransomware threats** through short-term engagements with our many **incident response partners**. However, we monitor far more customers full time than we do via IR engagements, and therefore, these ransomware incidents represent only a small fraction of our overall detection volume.

Interestingly, if we create a heatmap like the one above where we only include detections from our incident response work, we see a slightly different arrangement of techniques that does include impact tactics—as well as more defense evasion, more lateral movement, and less execution. This makes sense because in incident response engagements we are entering environments where a lot of the preliminary activity—the stuff we generally catch early for our full-time customers—has already occurred. In other words, we're already at the impactful part of the incident.

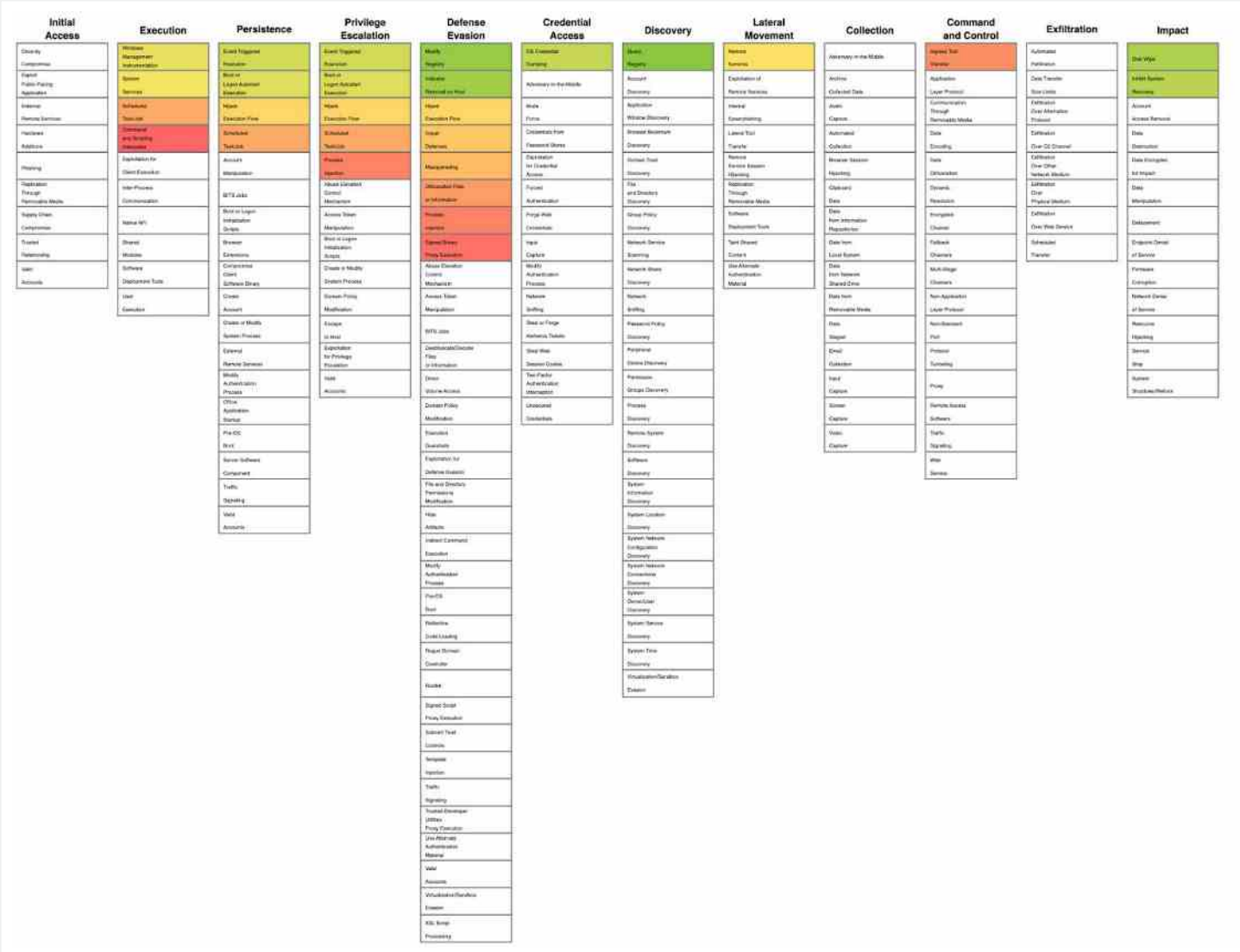


Figure 4: MITRE ATT&CK Navigator layer showing the 20 most prevalent ATT&CK techniques detected by Red Canary during incident response engagements

How to use our analysis

If your organization is able to follow the visibility, collection, and detection guidance in this report, you can effectively improve your defense-in-depth against the adversary actions that often lead to a serious incident. Of course, this is easier said than done. There are countless prerequisites to operationalizing this report, ranging from configuration challenges to developing plumbing that allows you to move telemetry from its source to its destination—whether that's a SIEM or some other aggregation point.

However, this analysis is still useful for practitioners or leaders who aren't immediately ready to operationalize it. For leaders, the most prevalent techniques can help you identify gaps as you develop a road map for improving coverage. You can assess your existing sources of collection against the ones listed in this report to inform your investments in new tools and personnel.

As a practitioner, you'll gain a better understanding of common adversary actions and what's likely to occur if an adversary gains access to your environment. You'll learn what malicious looks like in the form of telemetry and the many places you can look to find that telemetry. You'll gain familiarity with the principles of detection engineering by studying our detection opportunities. At a bare minimum, you and your team will be armed with hyper-relevant and easy-to-use **Atomic Red Team** tests that you can leverage to ensure that your existing security tooling does what you think it's supposed to do.

What's missing and why?

Red Canary is actively adopting new data sources that reach beyond the endpoint to enhance our detection, investigation, and incident handling capabilities, and you'll see evidence of this throughout the techniques section—particularly in the visibility, collection, and detection subsections. Even so, the majority of our detection analytics are based on endpoint telemetry and the majority of the endpoints we monitor are client workstations. This reality shapes our perspective and the contents of this report.

Given our vantage point and the defense-in-depth our detection analytics offer, we tend to detect the adversary behaviors that happen just after initial access. As a result, execution, privilege escalation, persistence, and defense evasion techniques are probably over-emphasized in our report. On the other hand, one of the most prevalent forms of initial access—email-based phishing—is under-represented. Under no circumstance should anyone interpret these findings to suggest that phishing protection is unimportant. To the contrary, phishing is among the most prevalent ways that adversaries initially access our customers' environments, and the data in this report does reflect a great number of

email-borne threats. However, Red Canary doesn't have as much early-stage visibility into phishing as we do into other techniques, which is precisely why this report works best as a complement to other reports from vendors with different vantage points—like those who make firewalls or email-monitoring products.

Further, discovery techniques are also underrepresented in this report. This is because discovery techniques can be incredibly noisy, generating prohibitively high volumes of false positives for our detection engineers. Beyond that, discovery-related alerts aren't always actionable since, for example, you can't really prevent someone from scanning a public-facing resource. This isn't to say we don't inform our customers of discovery activity. We absolutely do, but it's typically done manually by our detection engineers as they're analyzing potentially malicious events. Since our ATT&CK mapping happens at the detection-analytic level, prior to human analysis, the discovery activity isn't included in this report.

One final note: we overwhelmingly monitor Windows endpoints, and therefore we've included only limited information about macOS and Linux techniques in this section. To be very clear, we have robust detection coverage for macOS and Linux threats, but this report reflects the reality that Windows continues to dominate the enterprise marketplace.

Learn more about our top techniques

Conclusion

Thank you for devoting your time to absorbing this report; we appreciate your dedication to protecting your organization. We understand there are lots of reports floating around the information security community, and we take pride in our work to muffle the noise, opting instead for curated and actionable content. We hope the information encompassed in this report offers insights into how to improve your security posture and what you can do if you encounter any of the most prevalent threats, trends, and techniques. We will continue to update the **Red Canary blog** with relevant resources related to the Threat Detection Report and many other valuable resources you can use to take action.

If you have any questions or concerns, or just want to chat, please feel free to reach out to us at **info@redcanary.com**.

Contributors

Thank you to all our contributors who helped make this report possible.

Jimmy Astle
Kelsey Budnick
Susannah Clark
Aaron Dider
Brian Donohue
Jeff Felling
Thomas Gardner
Matt Graeber
Dominic Heidt
Dave Hull
Christina Johns
Jonny Johnson
Milan Klusacek
Tony Lambert
Adam Mashinchi
Keith McCammon
Paul Michaud
Katie Nickels
Lauren Podber
Justin Schoenfeld
Anna Seitz
Harrison Van Riper
Phillip Wells
Erin York


red canary