MITRE ENGENUITY | Center for Threat Informed Defense

Sightings Ecosystem: A Data-driven Analysis of ATT&CK in the Wild

Table of Contents

REPORT HIGHLIGHTS

Received 6m+ Sightings, pared down to 1.1m after normalizing data, across 184 unique techniques observed between April 2019 and July 2021.

2019 - 2021
APRIL JULY

6M+
SIGHTINGS

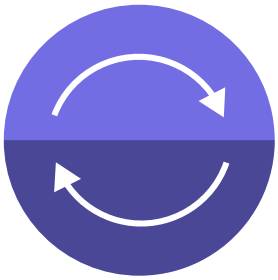
1.1M
NORMALIZED SIGHTINGS

184
UNIQUE TECHNIQUES

Provides a picture of

COMMON ADVERSARY BEHAVIORS

Which techniques adversaries use

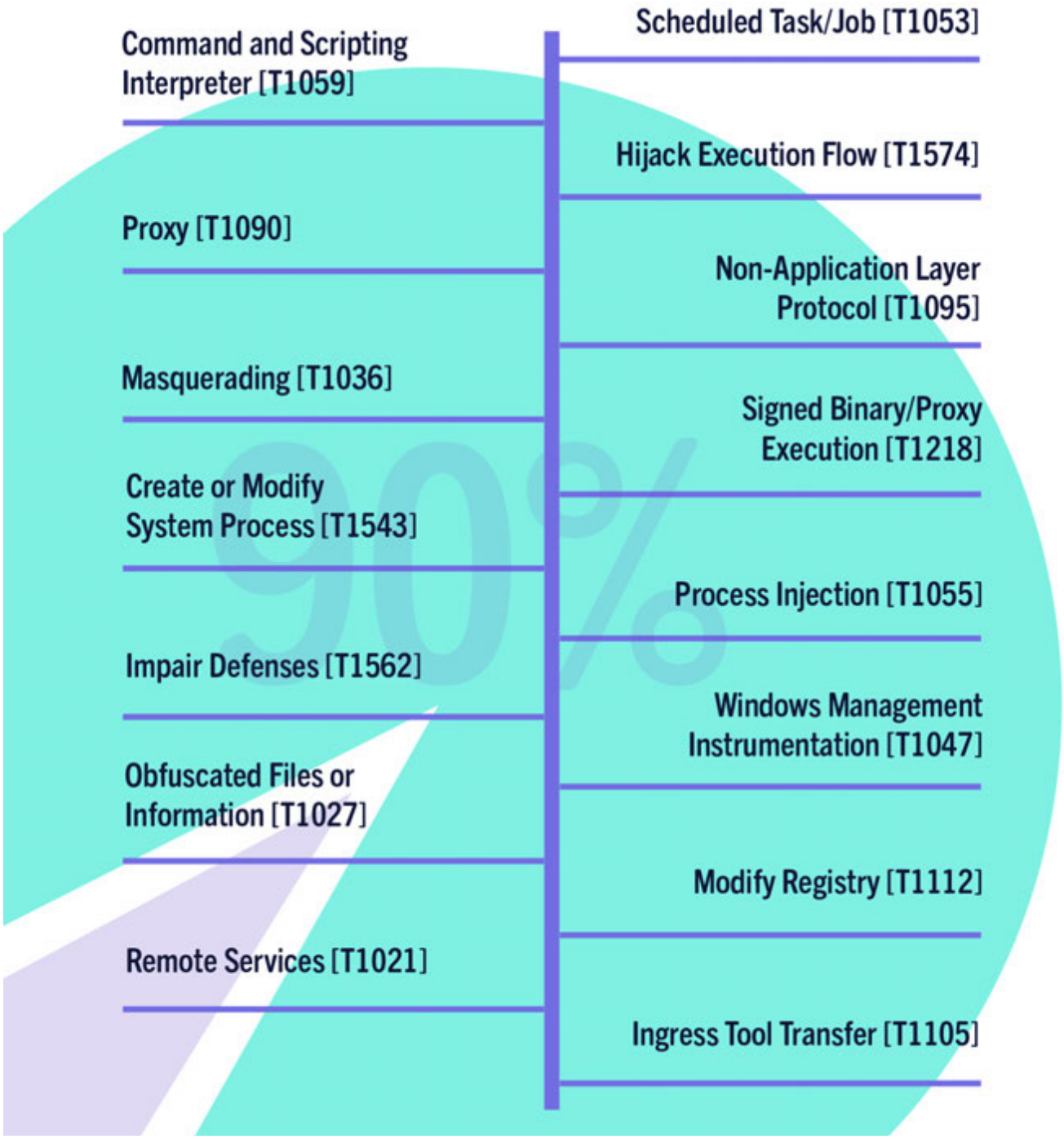


How their use changes over time

How adversaries use techniques together

15 TECHNIQUES

made up 90% of the observed techniques from April 2019 - July 2021. Most of these techniques abuse legitimate system tools.



10 NIST 800-53 CONTROLS

provide the coverage for the most observed techniques

1. SI-4 System Monitoring
2. CM-6 Configuration Settings
3. CM-2 Baseline Configuration
4. CM-7 Least Functionality
5. AC-3 Access Enforcement
6. AC-6 Least Privilege
7. AC-2 Account Management
8. AC-5 Separation of Duties
9. CM-5 Access Restrictions for Change
10. IA-2 Identification and Authentication (Organizational Users)

Introduction

The costs of intrusions are increasing as adversaries evolve their attacks. This evolution forces defenders to consider more and more attack sequences and techniques. Defending against all adversary scenarios is a big, and expensive, challenge. But how many of these techniques does the average organization need to be prepared for? Many known techniques are leveraged by state-sponsored groups or are used in targeted attacks but are not broadly used against opportunistic victims. For many of these organizations, the question becomes, "What techniques do we realistically need to be prepared to defend against?"

The Center for Threat-Informed Defense (Center), in collaboration with AttackIQ Inc., Fortinet's FortiGuard Labs, The Global Cyber Alliance, Verizon Business Solutions, and two other Center Participants, set out to answer these questions in the Sightings Ecosystem project. The Center recruited various security and service providers through existing connections, a blog post announcing the project, and the 2021 Black Hat cybersecurity event. Over several months, data contributors voluntarily shared over 6,000,000 Sightings of MITRE ATT&CK® techniques observed on their platforms. (A special thank you to ConnectWise Cyber Research Unit, LLC, FirstEnergy Corp, Red Canary, and our other data contributors. This effort would not have been possible without them.) This information created a picture of common adversary behavior including which techniques adversaries use, how their use changes over time, and how adversaries sequence techniques. Defenders can use this information to create a threat-informed defense against what they are most likely to see, not just the latest cyber threat headlines.

Our Findings

Fifteen techniques made up 90 percent of the techniques observed from 1 April 2019 to 31 July 2021. When looking at the 15 most observed techniques, one thing immediately stood out: most of these techniques abuse legitimate system tools. Reviewing the defense information for these techniques also underscores the idea that adversaries are attempting to appear as legitimate users. Therefore, creating strong baselines and restricting permissions is key to detecting and disrupting adversary behaviors.

We also found that techniques like *Scheduled Task/Job* [T1053] and *Command and Scripting Interpreter* [T1059] serve as facilitators for many of the other techniques. Building robust detections and logging around these keystone techniques can therefore significantly improve an organization's defense. Being able to identify adversary use of a keystone technique can help defenders identify other techniques used by adversaries in a victim environment.

Our data on adversaries provided some lessons for analysts and defenders. The collected data contained various irregularities, which we had to resolve before we could complete a trend analysis. This highlighted the importance of regularly reviewing collected data and verifying its integrity and the accuracy of the ATT&CK mappings. These lessons are described in greater detail in the Detections Lessons Learned section.

How can organizations use this information to defend themselves?

We provided references to open-source preventative controls and detections analytics for each of the top 15 observed techniques and sub-techniques. This can give organizations a low-cost, high-return starting point for building threat-informed defenses. Our analysis further identified which techniques we are seeing more of year over year, and which techniques we typically saw occur together. This information can further inform how defenders prioritize and hunt for adversary activity.

Organizations can review detections for these techniques and begin hunting for this activity within their environment. Because many of the most observed techniques are associated with legitimate system activities, real-time detection analytics may miss some adversaries' use of these techniques as they seek to avoid false positives. Proactively hunting for adversary presence allows network defenders to potentially uncover adversaries active in their network. After verifying no signs of malicious activity in their network, defenders can then establish a trusted baseline of normal activity from which to detect adversary behavior.

Sightings Ecosystem results can also be used to inform a security strategy. Defenders can use tools like MITRE's CALDERA platform, Red Canary's Atomic Red Team library, or the Center's Adversary Emulation Library to test defenses and detections on a recurring basis. These libraries contain tests for specific adversary behaviors observed in the Sightings dataset.

Defenders can further use Sightings data to assess how well their security products detect the most commonly observed techniques. Defenders can work with their security providers to populate an ATT&CK Navigator layer documenting all the techniques they are able to detect, as well as how they prioritize detecting them. ATT&CK Evaluations can also provide some insight on how well various products detect and prevent various techniques. Defenders can then compare their procured products' coverage to the most observed techniques. Any gaps in expected or needed coverage provide opportunities for improved defenses.

Framing Our Analysis

There are several important considerations when reading our results. First and foremost, we had a limited number of contributors. This means our data does not provide a comprehensive view of the threat landscape. There are techniques not present in our dataset which may be relevant to organizations depending on their environment and relative risk.

The data we received was limited to the visibility of the companies who graciously contributed their data to the Sightings Ecosystem. Each contributor has different visibility because of demographics of their customer base, the location of their sensor technology (e.g., external to the network or on an email server), and their relative ability to detect specific activity. We hoped to overcome these limitations by recruiting a large number of contributors, but our limited number means there remains a visibility bias in our results.

Our results are further limited by how our Contributors map techniques to ATT&CK. Depending on when techniques are mapped in an incident investigation and how formalized the mapping process is, it is not unrealistic to think that several Sightings may have been mis-mapped.

Aggregating data from multiple contributors also impacted our results. When we aggregated the data, we lost context on the adversaries and detections. We did not have deep insight into how the techniques are detected, which meant that we struggled to determine whether an increase in activity was caused by increased adversary activity or by improved detections. Throughout our results, we clarify these uncertainties and develop hypotheses which can be tested in future versions of the Sightings Ecosystem.

When looking at our data, it is important to remember that defending against our most observed techniques will not protect you from all adversary activity; it will only protect you from the adversary activity most observed by Sightings contributors. Despite these barriers, we were able to identify techniques that are frequently used by adversaries, and we provided guidance to defenders on concrete actions they can take.

What's in the Data

There are several important considerations when reading our results. First and foremost, we had a limited number of contributors. This means our data does not provide a comprehensive view of the threat landscape. There are techniques not present in our dataset which may be relevant to organizations depending on their environment and relative risk.

In the opening Sightings Ecosystem report, we collected over 1 million observed techniques across 800,000 Sightings encompassing 184 unique techniques and sub-techniques. Our analysis used version 9 of ATT&CK, released 29 April 2021. Most of the contributors were using earlier versions of ATT&CK, so we translated some data using the crosswalks provided by the ATT&CK team. This created a couple of issues with deprecated techniques. Where techniques were mapped to multiple sub-techniques, we identified the technique as the parent technique. To learn more about translating MITRE ATT&CK versions, review the blog post: https://medium.com/mitre-attack/attack-with-sub-techniques-is-now-just-attack-8fc20997d8de.

What is a Sighting? Each Sighting represents one or more ATT&CK techniques used by an adversary on (or to target) victim infrastructure. We also sought Sightings of malicious software and associated ATT&CK techniques, but we did not receive this data from all contributors and omitted the data from most of our analysis.

There were some concerns with the data. The dates from our dataset spanned several decades, which is not realistic to analyze because not all contributors provided the same time range. Additionally, we saw a disproportionate number of certain techniques. After investigating the root cause, we discovered that some of our contributors had either mislabeled or their platform was mis-tuned, causing them to identify certain techniques at a higher rate than others. To remove the impact of these mis-labeled techniques, we decided to limit our analysis to the timeframe after the mis-labeling/mis-tuning problem was resolved. Our resulting time frame for analysis was 1 April 2019 to 31 July 2021. This highlights the importance of data cleansing – detecting and deleting inaccurate records – and tuning of sensors. A complete understanding of data and systems is important for defending assets.

Data visualizations were created using Kibana Lens, Dash, and Microsoft Excel.

15 Most Observed Techniques

15 techniques accounted for 90 percent of the technique observations in our dataset. The most observed technique, *Scheduled Task/Job* [T1053], represented nearly a quarter of all Sightings. The remaining 108 techniques observed in our dataset only accounted for 10.3 percent of the technique Sightings.

The Top 15 Techniques:

1. Scheduled Task/Job [T1053]
2. Command and Scripting Interpreter [T1059]
3. Hijack Execution Flow [T1574]
4. Proxy [T1090]
5. Masquerading [T1036]
6. Signed Binary/Proxy Execution [T1218]
7. Create or Modify System Process [T1543]
8. Process Injection [T1055]
9. Impair Defenses [T1562]
10. Obfuscated Files or Information [T1027]
11. Remote Services [T1021]
12. Non-Application Layer Protocol [T1095]
13. Windows Management Instrumentation [T1047]
14. Modify Registry [T1112]
15. Ingress Tool Transfer [T1105]

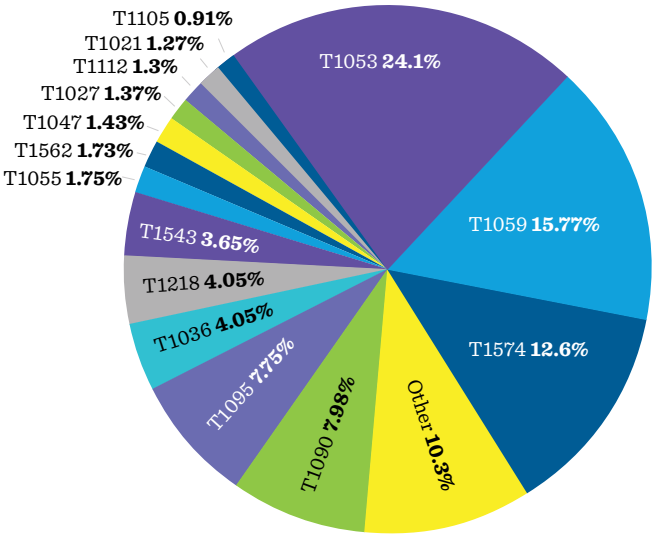


Figure 1 Total Dataset Technique Breakdown

Technique Sightings Details and Guidance for Defenders

Our top 15 observations changed when we added sub-techniques information. When we broke up the techniques, we found 26 techniques and sub-techniques accounted for 90 percent of the observations in our dataset. We opted to break up our analysis by the core technique and address relevant sub-techniques along the way below. By looking at each technique in-depth, including its observed sub-techniques, we can observe how frequently adversaries use these techniques, and understand how defenders can begin preventing and detecting each technique.

Reviewing the details of each observed technique revealed most of our dataset contains reporting on attacks impacting Windows systems, but attacks against Linux or macOS systems are also represented. We did not see techniques associated with any other ATT&CK matrices.

The details of each technique are explored below, alongside some prevention and detection guidance. This guidance can be further supplemented by reviewing the recommended mitigations and detections in the linked ATT&CK page. Prevention information is based on the Center's mapping between the National Institute of Standards and Technology (NIST) Special Publication (SP) 800-53 and ATT&CK techniques. Details on how the control relates to the technique are intended as a starting point, not as a complete implementation guide. NIST SP 800-53 provides more in-depth guidance about implementing and enhancing these controls within an environment.

Detection information was collected from MITRE's Cyber Analytics Repository and the SigmaHQ github page located here. The detections information is primarily for organizations with robust logging; reviewing the logsource field in some Sigma rules will allow organizations to determine if they have the appropriate logging to implement a rule. Organizations are strongly encouraged to test all rules in their environment before deploying to understand what false positives might result. We provided stable rules where we could, but many are listed as experimental. Testing them is therefore very important to verify the rules work and do not create a disrupting number of false positives.

1. Scheduled Task/Job [T1053]

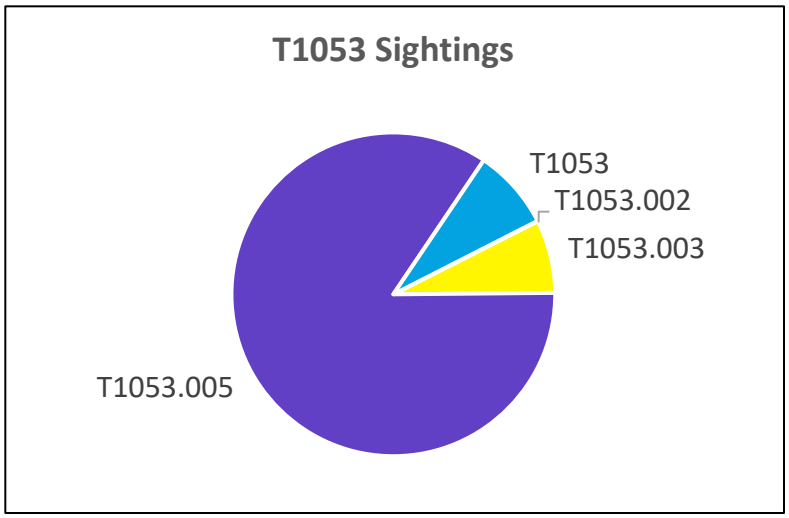


Figure 2 T1053 Technique Observation Breakdown

Scheduled Task [T1053.005] accounted for 85 percent of the *Scheduled Task/Job* [T1053] observations. *Cron* [T1053.003] further accounted for 7 percent of the *Scheduled Task/Job* [T1053] Sightings. Seeing both sub-techniques was interesting because it showed our dataset addressed both attacks against Windows systems (*Scheduled Task* [T1053.005]) and Linux and macOS systems (*Cron* [T1053.003]). Overall, we were not surprised to see the abundance of *Scheduled Task/Job* [T1053] observations. *Scheduled Task/Job* [T1053] and its various sub-techniques are often used to launch many other techniques. Additionally, as a narrowly defined activity with clear logging sources, detecting *Scheduled Task/Job* [T1053] can be easier than many other techniques.

Prevention

There are 14 security controls known to mitigate adversary use of Scheduled Task [T1053.005].

- AC-2 Account Management (Also mitigates Cron [T1053.003])
- AC-3 Access Enforcement (Also mitigates Cron [T1053.003])
- C-5 Separation of Duties (Also mitigates Cron [T1053.003])
- AC-6 Least Privilege (Also mitigates Cron [T1053.003])
- CA-8 Penetration Testing (Also mitigates Cron [T1053.003])
- CM-2 Baseline Configuration
- CM-5 Access Restrictions for Change (Also mitigates Cron [T1053.003])
- CM-6 Configuration Settings
- CM-7 Least Functionality
- CM-8 System Component Inventory
- IA-2 Identification and Authentication (organizational Users) (Also mitigates Cron [T1053.003])
- IA-4 Identifier Management
- RA-5 Vulnerability Monitoring and Scanning (Also mitigates Cron [T1053.003])
- SI-4 System Monitoring (Also mitigates Cron [T1053.003])

Detection

There are multiple published Sigma rules for detecting *Scheduled Task* [T1053.005] and *Cron* [T1053.003]. Legitimate administration activities and software installation can result in false positives for many of these rules. Before implementing any of these rules, test them in an environment and consider creating policies for identification of false positives.

CAR contains analytics for two of the sub-techniques observed in our dataset: *Scheduled Task* [T1053.005] and *At (Windows)* [T1053.002]. Because *At (Windows)* [T1053.002] accounted for less than a ten thousandth of a percentage of overall *Scheduled Task/Job* [T1053] observations, we have only provided the analytics for *Scheduled Task* [T1053.005].

Rules for detecting *Scheduled Task* [T1053.005]:

- CAR-2013-01-002: Autorun Differences
- CAR-2013-04-002: Quick execution of a series of suspicious commands
- CAR-2013-08-001: Execution with schtasks
- CAR-2015-04-002: Remotely Scheduled Tasks via Schtasks
- CAR-2020-09-001: Scheduled Task - FileAccess
- https://github.com/SigmaHQ/sigma/blob/77c8225db37099adc3b27513039979b7a63281ca/rules/windows/registry_event/sysmon_taskcache_entry.yml
- https://github.com/SigmaHQ/sigma/blob/08ca62cc8860f4660e945805d0dd615ce75258c1/rules/windows/process_creation/win_susp_schtask_creation_temp_folder.yml
- https://github.com/SigmaHQ/sigma/blob/59000b993d6280d9bf063eefdcdf30ea0e83aa5e/rules/windows/process_creation/win_susp_schtask_creation.yml

Rules for detecting *Cron* [T1053.003]:

- https://github.com/SigmaHQ/sigma/blob/08ca62cc8860f4660e945805d0dd615ce75258c1/rules/linux/macos_schedule_task_job_cron.yml
- https://github.com/SigmaHQ/sigma/blob/08ca62cc8860f4660e945805d0dd615ce75258c1/rules/linux/lnx_schedule_task_job_cron.yml
- https://github.com/SigmaHQ/sigma/blob/8595478b360c48c3160cb8ebe9ae403802524ef0/rules/linux/file_create/cron_files.yml *This rule alerts on any cron file creation and can result in many false positives if cron jobs are frequently created in an environment.

2. Command and Scripting Interpreter [T1059]

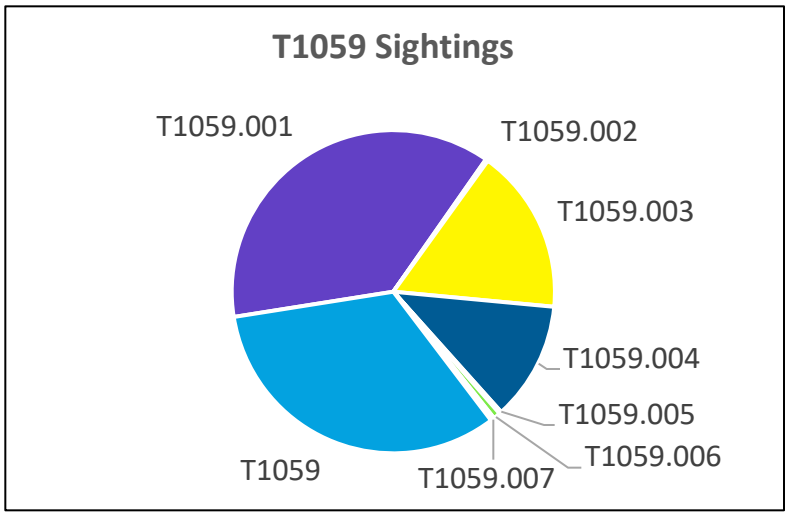


Figure 3 T1059 Technique Observation Breakdown

More than a third of *Command and Scripting Interpreter* [T1059] observations were of *PowerShell* [T1059.001]. *Windows Command Shell* [T1059.003] and *Unix Shell* [T1059.004] further accounted for more than a quarter of the observations. *PowerShell* [T1059.001] and *Windows Command Shell* [T1059.003] are both sub-techniques used on Windows systems. *Unix Shell* [T1059.004] can be used on Linux and macOS systems, further supporting the conclusion from *Scheduled Task/Job* [T1053] that our dataset includes more than just attacks targeting Windows systems. As with *Scheduled Task/Job* [T1053], we expected to see a large number of observations of *Command and Scripting Interpreter* [T1059] because adversaries frequently use this technique to launch other techniques. Detecting its use, however, can be little more complicated because these portals are regularly used for legitimate activity. Legitimate systems administration activities may trigger false positives for many of the analytics provided below.

Prevention

There are 16 security controls known to mitigate adversary use of *PowerShell* [T1059.001].

- AC-2 Account Management
- AC-3 Access Enforcement
- AC-5 Separation of Duties
- AC-6 Least Privilege
- CM-2 Baseline Configuration
- CM-5 Access Restrictions for Change
- CM-6 Configuration Settings
- CM-8 System Component Inventory
- IA-2 Identification and Authentication (organizational Users)
- IA-8 Identification and Authentication (non-organizational Users)
- IA-9 Service Identification and Authentication
- RA-5 Vulnerability Monitoring and Scanning
- SI-2 Flaw Remediation
- SI-3 Malicious Code Protection

- SI-4 System Monitoring (Also mitigates *Windows Command Shell* [T1059.003] and *Unix Shell* [T1059.004])
- SI-7 Software, Firmware, and Information Integrity (Also mitigates *Windows Command Shell* [T1059.003] and *Unix Shell* [T1059.004])

Windows *Command Shell* [T1059.003] and *Unix Shell* [T1059.004] are further mitigated by two controls not applicable to *PowerShell* [T1059.001].

- CM-7 Least Functionality
- SI-10 Information Input Validation

Detection

Various Sigma rules exist for this technique and its sub-techniques. The list of detections below is not comprehensive and is intended as a starting point. CAR also contains multiple analytics for *PowerShell* [T1059.001] and *Windows Command Shell* [T1059.003] which are also provided below.

Rules for detecting *PowerShell* [T1059.001]:

- CAR-2014-04-003: Powershell Execution
- CAR-2014-11-004: Remote PowerShell Sessions
- https://github.com/SigmaHQ/sigma/blob/master/rules/windows/process_creation/win_susp_powershell_parent_combo.yml *Microsoft Operations Manager and certain other scripts may result in false positives for this rule.
- https://github.com/SigmaHQ/sigma/blob/master/rules/windows/process_creation/win_powershell_download.yml
- https://github.com/SigmaHQ/sigma/blob/master/rules/windows/process_creation/win_susp_powershell_enc_cmd.yml

Rules for detecting *Windows Command Shell* [T1059.003]:

- CAR-2013-02-003: Processes Spawning cmd.exe
- CAR-2014-11-002: Outlier Parents of Cmd
- https://github.com/SigmaHQ/sigma/blob/08ca62cc8860f4660e945805d0dd615ce75258c1/rules/windows/process_creation/win_commandline_path_traversal.yml
- https://github.com/SigmaHQ/sigma/blob/08ca62cc8860f4660e945805d0dd615ce75258c1/rules/windows/process_creation/win_susp_cmd_http_appdata.yml

Rules for detecting *Unix Shell* [T1059.004]:

- https://github.com/SigmaHQ/sigma/blob/08ca62cc8860f4660e945805d0dd615ce75258c1/rules/linux/auditd/lnx_auditd_susp_cmds.yml *This rule may result in false positives arising from administrative activity that use the same commands.
- https://github.com/SigmaHQ/sigma/blob/08ca62cc8860f4660e945805d0dd615ce75258c1/rules/linux/lnx_shell_susp_rev_shells.yml

3. Hijack Execution Flow [T1574]

Nearly every *Hijack Execution Flow* [T1574] observation was for *DLL Search Order Hijacking* [T1574.001]. Only 12 other observations of this technique were associated with a different sub-technique.

Prevention

There are nine security controls known to help prevent *DLL Search Order Hijacking* [T1574.001].

- CA-8 Penetration Testing
- CM-2 Baseline Configuration
- CM-6 Configuration Settings
- CM-7 Least Functionality
- RA-5 Vulnerability Monitoring and Scanning
- SI-10 Information Input Validation
- SI-3 Malicious Code Protection
- SI-4 System Monitoring
- SI-7 Software, Firmware, and Information Integrity

Detection

- https://github.com/SigmaHQ/sigma/blob/08ca62cc8860f4660e945805d0dd615ce75258c1/rules/windows/image_load/sysmon_svchost_dll_search_order_hijack.yml
- https://github.com/SigmaHQ/sigma/blob/e849af9df0129cb496f4701751dd93cfb5284849/rules/windows/file_delete/win_cve_2021_1675_printspooler_del.yml *This rule is experimental and may result in false positives.

4. Proxy [T1090]

Only 37 observations of this technique were for something other than the core technique.

Prevention

Twelve security controls can help mitigate adversary use of *Proxy* [T1090].

- AC-3 Access Enforcement
- AC-4 Information Flow Enforcement
- CA-7 Continuous Monitoring
- CM-2 Baseline Configuration
- CM-6 Configuration Settings
- CM-7 Least Functionality
- SC-7 Boundary Protection
- SC-8 Transmission Confidentiality and Integrity
- SI-10 Information Input Validation
- SI-15 Information Output Filtering
- SI-3 Malicious Code Protection
- SI-4 System Monitoring

Detection

https://github.com/SigmaHQ/sigma/blob/08ca62cc8860f4660e945805d0dd615ce75258c1/rules/windows/process_creation/win_netsh_port_fwd_3389.yml *This rule may result in false positives arising from administrative activity that use the same commands.

5. Non-Application Layer Protocol [T1095]

There are no identified sub-techniques for *Non-Application Layer Protocol* [T1095] because the list of potential protocols adversaries can use is extensive.

Prevention

Eleven controls can help mitigate adversary use of *Non-Application Layer Protocol* [T1095].

- AC-3 Access Enforcement
- AC-4 Information Flow Enforcement
- CA-7 Continuous Monitoring
- CM-2 Baseline Configuration
- CM-6 Configuration Settings
- CM-7 Least Functionality
- SC-7 Boundary Protection
- SI-10 Information Input Validation
- SI-15 Information Output Filtering
- SI-3 Malicious Code Protection
- SI-4 System Monitoring

Detection

Non-Application Layer Protocol [T1095] can include the use of many different protocols. Therefore, analytics for detecting this technique can be varied. The analytics below are for two tools adversaries can use to define the protocol then used for *Command and Control* [TA0011] communications.

- https://github.com/SigmaHQ/sigma/blob/e5b3a1cc14aaad6f2acc569fab9849567f98df3e/rules/windows/powershell/powershell_module/powershell_powercat.yml
- https://github.com/SigmaHQ/sigma/blob/5b95ef0872a52fb173bbb2146a65f9d16215f09e/rules/windows/process_creation/sysmon_netcat_execution.yml *Legitimate ncat use will cause false positives with this rule.

6. Masquerading [T1036]

Match Legitimate Name or Location [T1036.005] accounted for the majority of the *Masquerading* [T1036] observations. The other notable sub-technique observed was *Rename System Utilities* [T1036.003].

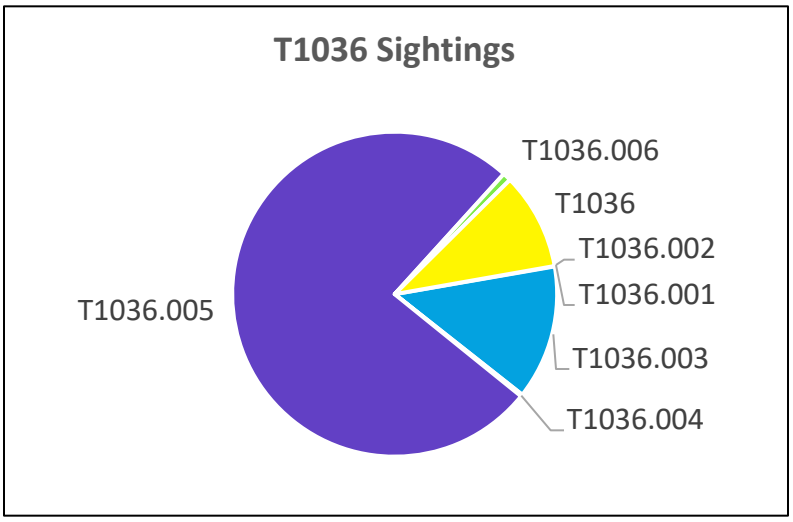


Figure 4 Technique Observation Breakdown

Prevention

There are 12 security controls that will help mitigate adversary use of *Match Legitimate Name or Location* [T1036.005].

- AC-2 Account Management (Also mitigates *Rename System Utilities* [T1036.003])
- AC-3 Access Enforcement (Also mitigates *Rename System Utilities* [T1036.003])
- AC-6 Least Privilege (Also mitigates *Rename System Utilities* [T1036.003])
- CA-7 Continuous Monitoring (Also mitigates *Rename System Utilities* [T1036.003])
- CM-2 Baseline Configuration (Also mitigates *Rename System Utilities* [T1036.003])
- CM-6 Configuration Settings (Also mitigates *Rename System Utilities* [T1036.003])
- CM-7 Least Functionality
- IA-9 Service Identification and Authentication
- SI-10 Information Input Validation
- SI-3 Malicious Code Protection (Also mitigates *Rename System Utilities* [T1036.003])
- SI-4 System Monitoring (Also mitigates *Rename System Utilities* [T1036.003])
- SI-7 Software, Firmware, and Information Integrity

Detection

- CAR-2013-05-002: Suspicious Run Locations
- https://github.com/SigmaHQ/sigma/blob/08ca62cc8860f4660e945805d0dd615ce75258c1/rules/windows/process_creation/win_susp_msiexec_cwd.yml
- https://github.com/SigmaHQ/sigma/blob/08ca62cc8860f4660e945805d0dd615ce75258c1/rules/windows/process_creation/win_renamed_binary_highly_relevant.yml
- https://github.com/SigmaHQ/sigma/blob/08ca62cc8860f4660e945805d0dd615ce75258c1/rules/windows/process_creation/win_proc_wrong_parent.yml *Some security products may cause false positives for this rule.
- https://github.com/SigmaHQ/sigma/blob/08ca62cc8860f4660e945805d0dd615ce75258c1/rules/windows/process_creation/win_renamed_binary.yml

7. Signed Binary/Proxy Execution [T1218]

Mshta [T1218.005] and *Rundll32* [T1218.011] were the most frequently observed *Signed Binary/Proxy Execution* [T1218] sub-techniques, followed by *Regsvr32* [T1218.010] and *CMSTP* [T1218.003]. These sub-techniques are all frequently used to accomplish other techniques because they enable adversaries to execute code outside of standard pathways more easily detected by security tools.

Prevention

The same security controls can help mitigate adversary use of *Mshta* [T1218.005] and *CMSTP* [T1218.003].

- CM-2 Baseline Configuration
- CM-6 Configuration Settings
- CM-7 Least Functionality
- CM-8 System Component Inventory
- RA-5 Vulnerability Monitoring and Scanning
- SI-10 Information Input Validation
- SI-4 System Monitoring
- SI-7 Software, Firmware, and Information Integrity

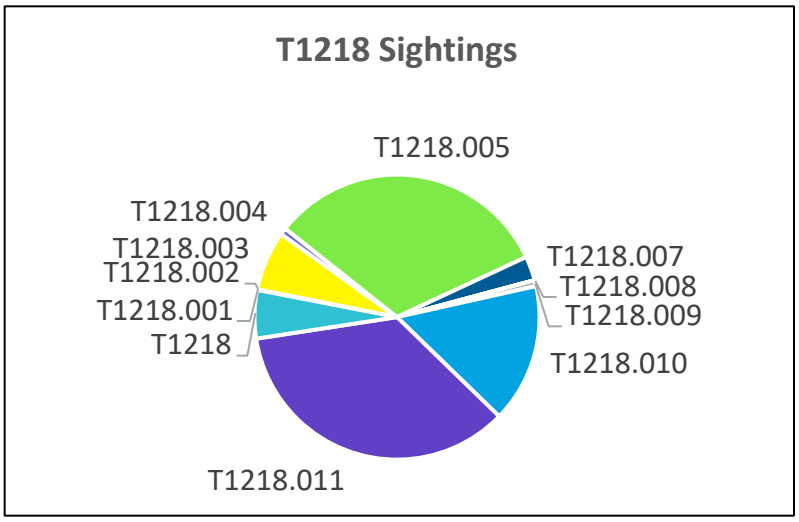


Figure 5 Technique Observation Breakdown

The same security controls can help mitigate adversary use of *Rundll32* [T1218.011] and *Regsvr32* [T1218.010].

- CA-7 Continuous Monitoring
- SI-10 Information Input Validation
- SI-4 System Monitoring
- SI-7 Software, Firmware, and Information Integrity Detection

Detection

Rules for detecting *Mshta* [T1218.005]:

- https://github.com/SigmaHQ/sigma/blob/ff0f1a0222b5100120ae3e43df18593f904c69c0/rules/windows/process_creation/win_mshta_javascript.yml
- https://github.com/SigmaHQ/sigma/blob/08ca62cc8860f4660e945805d0dd615ce75258c1/rules/windows/process_creation/win_mshta_spawn_shell.yml

Rules for detecting *Rundll32* [T1218.011]:

- CAR-2014-03-006: RunDLL32.exe monitoring
- https://github.com/SigmaHQ/sigma/blob/7288ae93b9ec8d09f56cdc623a44a21fa0826afb/rules/windows/process_creation/process_creation_cobaltstrike_load_by_rundll32.yml

- https://github.com/SigmaHQ/sigma/blob/08ca62cc8860f4660e945805d0dd615ce75258c1/rules/windows/process_creation/win_susp_rundll32_sys.yml
- https://github.com/SigmaHQ/sigma/blob/08ca62cc8860f4660e945805d0dd615ce75258c1/rules/windows/process_creation/win_susp_control_dll_load.yml

Rules for detecting Regsvr32 [T1218.010]:

- CAR-2019-04-002: Generic Regsvr32
- CAR-2019-04-003: Squiblydoo
- https://github.com/SigmaHQ/sigma/blob/6d56e400d209daa77a7900d950a7c587dc0cd2e5/rules/windows/network_connection/sysmon_regsvr32_network_activity.yml
- https://github.com/SigmaHQ/sigma/blob/3107ede1c4d253c89a26f3a0be79122a3a562f29/rules/windows/process_creation/process_creation_lolbins_by_office_applications.yml
  - This rule specifically looks for new living-off-the-land binaries (LOLBins) and can be customized depending on the LOLBins you are concerned about in your environment. The most complete listing of LOLBins is provided by the LOLBAS project here.

Rules for detecting *CMSTP* [T1218.003]:

- CAR-2020-11-010: CMSTP
- https://github.com/SigmaHQ/sigma/blob/6d0d58dfe240f7ef46e7da928c0b65223a46c3b2/rules/windows/registry_event/sysmon_cmstp_execution_by_registry.yml
- https://github.com/SigmaHQ/sigma/blob/6d0d58dfe240f7ef46e7da928c0b65223a46c3b2/rules/windows/process_access/sysmon_cmstp_execution_by_access.yml
- Legitimate CMSTP use will cause false positives for both rules, but that activity is uncommon in many environments.

8. Create or Modify System Process [T1543]

Most *Create or Modify System Process* [T1543] observations were of *Windows Service* [T1543.003]. One common adversary use of this technique involves using *Masquerading* [T1036], another top technique.

Prevention

There are 15 security controls that help mitigate adversary use of *Windows Service* [T1543.003].

- AC-17 Remote Access
- AC-2 Account Management
- AC-3 Access Enforcement
- AC-5 Separation of Duties
- AC-6 Least Privilege
- CA-8 Penetration Testing
- CM-11 User-installed Software
- CM-2 Baseline Configuration
- CM-5 Access Restrictions for Change
- CM-6 Configuration Settings
- CM-7 Least Functionality
- IA-2 Identification and Authentication (organizational Users)
- IA-4 Identifier Management
- RA-5 Vulnerability Monitoring and Scanning
- SI-4 System Monitoring

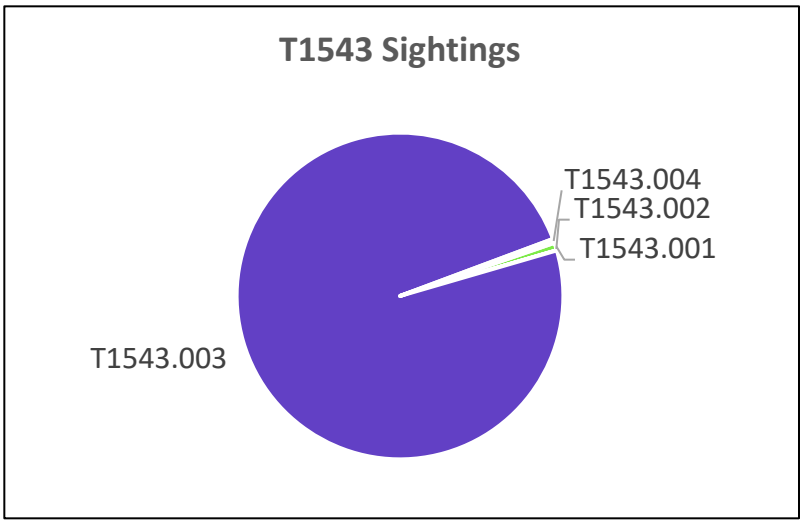


Figure 6 Technique Observation Breakdown

Detection

- CAR-2013-01-002: Autorun Differences
- CAR-2013-04-002: Quick execution of a series of suspicious commands
- CAR-2013-09-005: Service Outlier Executables
- CAR-2014-02-001: Service Binary Modifications
- CAR-2014-03-005: Remotely Launched Executables via Services
- CAR-2014-05-002: Services launching Cmd
- https://github.com/SigmaHQ/sigma/blob/ff0f1a0222b5100120ae3e43df18593f904c69c0/rules/windows/process_creation/win_new_service_creation.yml
- https://github.com/SigmaHQ/sigma/blob/71625c54f0469b6c8edad6fb1a3c23dcf17687e5/rules/windows/builtin/win_susp_proceshacker.yml *Legitimate services creation will trigger a false positive for this rule.

9. Process Injection [T1055]

Most *Process Injection* [T1055] observations were for the core technique, but more than 10 percent were of *Process Hollowing* [T1055.012].

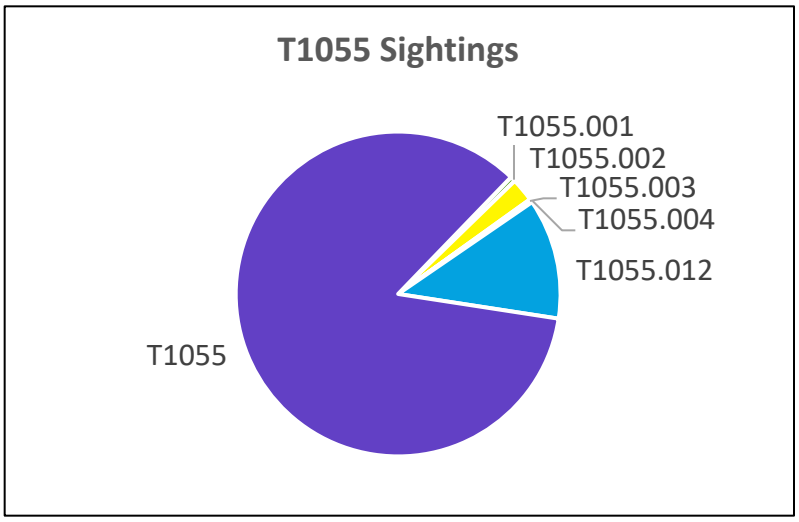


Figure 7 Technique Observation Breakdown

Prevention

There are 12 security controls known to help mitigate adversary use of *Process Injection* [T1055]. Six of these controls also help mitigate adversary use of *Process Hollowing* [T1055.012].

- AC-2 Account Management
- AC-3 Access Enforcement
- AC-5 Separation of Duties
- AC-6 Least Privilege (Also mitigates *Process Hollowing* [T1055.012])
- CM-5 Access Restrictions for Change
- CM-6 Configuration Settings
- IA-2 Identification and Authentication (organizational Users)
- SC-18 Mobile Code (Also mitigates *Process Hollowing* [T1055.012])
- SC-7 Boundary Protection (Also mitigates *Process Hollowing* [T1055.012])
- SI-2 Flaw Remediation (Also mitigates *Process Hollowing* [T1055.012])
- SI-3 Malicious Code Protection (Also mitigates *Process Hollowing* [T1055.012])
- SI-4 System Monitoring (Also mitigates *Process Hollowing* [T1055.012])

Detection

Adversaries can use multiple means to accomplish *Process Injection* [T1055], so these analytics are not comprehensive. These analytics are focused on detecting if a process is performing unusual actions, versus detecting the actual mechanism of how the adversary accomplishes *Process Injection* [T1055]. Organizations with strong baselines of expected behavior can use these analytics to generate custom rules based on expected process actions in their environment.

Rules for detecting *Process Injection* [T1055]:

- https://github.com/SigmaHQ/sigma/blob/8aabb58eca06cc44ae21ae4d091793d8c5ca6a23/rules/windows/create_remote_thread/sysmon_createremotethread_loadlibrary.yml
- https://github.com/SigmaHQ/sigma/blob/b267504708d7093bc940d96de35f5e5c99b2aa8f/rules/windows/process_access/sysmon_in_memory_assembly_execution.yml
- https://github.com/SigmaHQ/sigma/blob/08ca62cc8860f4660e945805d0dd615ce75258c1/rules/windows/process_creation/win_mavinject_proc_inj.yml

Rules for detecting *Process Hollowing* [T1055.012]:

- CAR-2020-11-004: Processes Started From Irregular Parent
- https://github.com/SigmaHQ/sigma/blob/1ff5e226ad8bed34916c16ccc77ba281ca3203ae/rules/windows/create_remote_thread/sysmon_cactustorch.yml

10. Impair Defenses [T1562]

Nearly all *Impair Defenses* [T1562] observations were of *Disable or Modify Tools* [T1562.001].

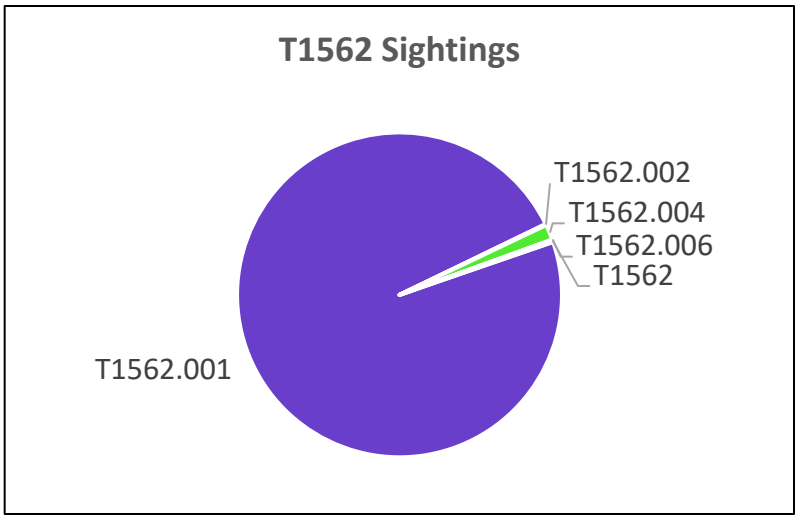


Figure 8 Technique Observation Breakdown

Prevention

There are 13 controls that can help mitigate adversary attempts to *Disable or Modify Tools* [T1562.001].

- AC-2 Account Management
- AC-3 Access Enforcement
- AC-5 Separation of Duties
- AC-6 Least Privilege
- CA-7 Continuous Monitoring
- CM-2 Baseline Configuration
- CM-5 Access Restrictions for Change
- CM-6 Configuration Settings
- CM-7 Least Functionality
- IA-2 Identification and Authentication (organizational Users)
- SI-3 Malicious Code Protection
- SI-4 System Monitoring
- SI-7 Software, Firmware, and Information Integrity

Detection

- There are many rules for *Disable or Modify Tools* [T1562.001], depending on which tools are being disabled. The rules provided below can serve as a starting point and are not comprehensive. Organizations may want to consider creating rules to detect attempts to disable security tools used in their environment.
- CAR-2013-04-002: Quick execution of a series of suspicious commands
- CAR-2016-04-003: User Activity from Stopping Windows Defensive Services
- CAR-2021-01-007: Detecting Tampering of Windows Defender Command Prompt
- https://github.com/SigmaHQ/sigma/blob/5951ad1d9a781a49d61df9af03c7b83ac67a0012/rules/windows/other/win_defender_disabled.yml
- https://github.com/SigmaHQ/sigma/blob/682e0458a336c3a6e93b18f7e972e1d67ef01598/rules/windows/process_creation/sysmon_uninstall_crowdstrike_falcon.yml
- https://github.com/SigmaHQ/sigma/blob/321c31cb7b1cc2c11d1c643490c78d1d83fbfc12/rules/windows/registry_event/sysmon_removal_amsi_registry_key.yml

11. Windows Management Instrumentation [T1047]

Windows Management Instrumentation [T1047] does not have any associated sub-techniques. Adversaries can use this technique locally or remotely to execute malicious commands and payload. When used remotely, *Windows Management Instrumentation* [T1047] can be facilitated by Remote Services [T1021], particularly *Windows Remote Management* [T1021.006]. This technique and sub-technique are also part of the top 15 most observed techniques.

Prevention

Eight controls can help mitigate adversary use of *Windows Management Instrumentation* [T1047].

- AC-17 Remote Access
- AC-2 Account Management
- AC-3 Access Enforcement
- AC-5 Separation of Duties
- AC-6 Least Privilege
- CM-5 Access Restrictions for Change
- CM-6 Configuration Settings
- IA-2 Identification and Authentication (organizational Users)

Detection

Windows Management Instrumentation [T1047] is a legitimate tool, so the analytics below focus on either whether the tool was launched remotely or suspicious tasks implemented using *Windows Management Instrumentation* [T1047]. Legitimate use of *Windows Management Instrumentation* [T1047] which meets these criteria will trigger false positives.

- CAR-2014-11-007: Remote Windows Management Instrumentation (WMI) over RPC
- CAR-2014-12-001: Remotely Launched Executables via WMI
- https://github.com/SigmaHQ/sigma/blob/08ca62cc8860f4660e945805d0dd615ce75258c1/rules/windows/builtin/win_susp_wmi_login.yml *This rule detects logins performed with Windows Management Instrumentation so it may result in false positives from legitimate system administration.
- https://github.com/SigmaHQ/sigma/blob/08ca62cc8860f4660e945805d0dd615ce75258c1/rules/windows/process_creation/win_susp_wmic_proc_create_rundll32.yml

12. Obfuscated Files or Information [T1027]

The core technique accounted for most observations of *Obfuscated Files or Information* [T1027]. We expected to see adversaries use *Deobfuscate/Decode Files or Information* [T1140] at a similar rate as *Obfuscated Files or Information* [T1027], but instead *Obfuscated Files or Information* [T1027] was observed three times more often than *Deobfuscate/Decode Files or Information* [T1140]. This highlights the important of detecting and preventing adversary use of *Obfuscated Files or Information* [T1027] rather than waiting to detect files or information until after they are deobfuscated or decoded.

Prevention

There are four controls that can help mitigate Obfuscated Files or Information [T1027].

- SI-2 Flaw Remediation
- SI-3 Malicious Code Protection
- SI-4 System Monitoring
- SI-7 Software, Firmware, and Information Integrity

Detection

- https://github.com/SigmaHQ/sigma/blob/08ca62cc8860f4660e945805d0dd615ce75258c1/rules/windows/process_creation/win_powershell_b64_shellcode.yml
- https://github.com/SigmaHQ/sigma/blob/3107ede1c4d253c89a26f3a0be79122a3a562f29/rules/windows/process_creation/win_invoke_obfuscation_via_var.yml
- https://github.com/SigmaHQ/sigma/blob/08ca62cc8860f4660e945805d0dd615ce75258c1/rules/linux/lnx_base64_decode.yml
- https://github.com/SigmaHQ/sigma/blob/08ca62cc8860f4660e945805d0dd615ce75258c1/rules/linux/macos_base64_decode.yml

13. Modify Registry [T1112]

Modify Registry [T1112] does not have any associated sub-techniques. The technique often requires *Valid Accounts* [T1078], but this technique was not observed in the top 15 techniques. In fact, Valid Accounts [T1078] was observed a little more than 100 times in our dataset. This discrepancy reflects either an issue in mappings of Valid Accounts [T1078] or that adversaries use *Modify Registry* [T1112] differently than it is communicated on its ATT&CK page.

Prevention

There are two controls which can help mitigate adversary attempts to *Modify Registry* [T1112].

- AC-6 Least Privilege
- CM-7 Least Functionality

Detection

- CAR-2013-03-001: Reg.exe called from Command Shell
- CAR-2014-11-005: Remote Registry
- CAR-2020-05-003: Rare LolBAS Command Lines
- https://github.com/SigmaHQ/sigma/blob/b08b3e2b0d5111c637dbede1381b07cb79f8c2eb/rules/windows/malware/registry_event_mal_ursnif.yml
- https://github.com/SigmaHQ/sigma/blob/b08b3e2b0d5111c637dbede1381b07cb79f8c2eb/rules/windows/malware/registry_event_mal_azorult.yml
- https://github.com/SigmaHQ/sigma/blob/e9260679d4aeae7f696001c5b14d318d31c8f076/rules/windows/registry_event/registry_event_net_ntlm_downgrade.yml
- https://github.com/SigmaHQ/sigma/blob/b08b3e2b0d5111c637dbede1381b07cb79f8c2eb/rules/windows/malware/registry_event_mal_blue_mockingbird.yml

14. Remote Services [T1021]

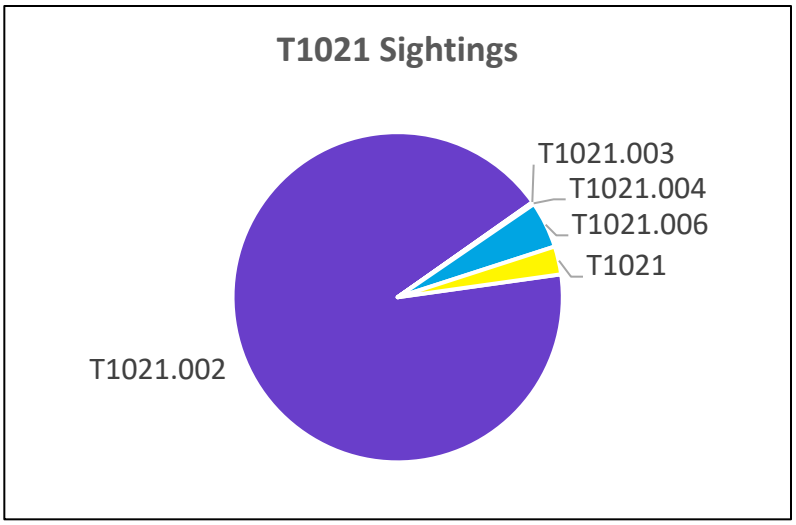


Figure 9 Technique Observation Breakdown

Most observations were of *SMB/Windows Admin Shares* [T1021.002], but there were also observations of *Windows Remote Management* [T1021.006] and the core technique *Remote Services* [T1021]. Adversaries reportedly often use *Valid Accounts* [T1078] to interact with *SMB/Windows Admin Shares* [T1021.002] and *Windows Remote Management* [T1021.006], but that association was not supported in our dataset. We observed *Valid Accounts* [T1078] a little more than 100 times, indicating either a potential issue with how that technique is captured or that adversaries use these techniques differently than they are characterized in the ATT&CK references. *Windows Remote Management* [T1021.006] can be used to remotely interact with *Windows Management Instrumentation* [T1047], another top technique.

Prevention

There are 16 security controls that can mitigate adversary use of *SMB/Windows Admin Shares* [T1021.002]. Thirteen of these controls can also help mitigate adversary use of *Windows Remote Management* [T1021.006].

- AC-17 Remote Access (Also mitigates adversary use of *Windows Remote Management* [T1021.006])
- AC-2 Account Management (Also mitigates adversary use of *Windows Remote Management* [T1021.006])
- AC-3 Access Enforcement (Also mitigates adversary use of *Windows Remote Management* [T1021.006])
- AC-4 Information Flow Enforcement (Also mitigates adversary use of *Windows Remote Management* [T1021.006])
- AC-5 Separation of Duties (Also mitigates adversary use of *Windows Remote Management* [T1021.006])
- AC-6 Least Privilege (Also mitigates adversary use of *Windows Remote Management* [T1021.006])
- CA-7 Continuous Monitoring
- CM-2 Baseline Configuration (Also mitigates adversary use of *Windows Remote Management* [T1021.006])
- CM-5 Access Restrictions for Change (Also mitigates adversary use of *Windows Remote Management* [T1021.006])
- CM-6 Configuration Settings (Also mitigates adversary use of *Windows Remote Management* [T1021.006])

- CM-7 Least Functionality (Also mitigates adversary use of *Windows Remote Management* [T1021.006])
- IA-2 Identification and Authentication (organizational Users) (Also mitigates adversary use of *Windows Remote Management* [T1021.006])
- SC-7 Boundary Protection (Also mitigates adversary use of *Windows Remote Management* [T1021.006])
- SI-10 Information Input Validation
- SI-15 Information Output Filtering
- SI-4 System Monitoring (Also mitigates adversary use of *Windows Remote Management* [T1021.006])

In addition to the above controls, adversary use of *Windows Remote Management* [T1021.006] can be mitigated by three other controls.

- CM-8 System Component Inventory
- RA-5 Vulnerability Monitoring and Scanning
- SC-46 Cross Domain Policy Enforcement

Detection

- CAR-2013-01-003: SMB Events Monitoring
- CAR-2013-04-002: Quick execution of a series of suspicious commands
- CAR-2013-05-003: SMB Write Request
- CAR-2013-05-005: SMB Copy and Execution
- CAR-2014-05-001: RPC Activity
- https://github.com/SigmaHQ/sigma/blob/8aabb58eca06cc44ae21ae4d091793d8c5ca6a23/rules/windows/builtin/win_protected_storage_service_access.yml
- https://github.com/SigmaHQ/sigma/blob/08ca62cc8860f4660e945805d0dd615ce75258c1/rules/windows/process_creation/win_rundll32_without_parameters.yml

15. Ingress Tool Transfer [T1105]

Ingress Tool Transfer [T1105] does not have any associated sub-techniques. It may be accomplished through *Command and Control* [TA0011] channels on Windows systems or through native tools on Mac and Linux systems.

Prevention

Eight security controls can help prevent adversary use of *Ingress Tool Transfer* [T1105].

- AC-4 Information Flow Enforcement
- CA-7 Continuous Monitoring
- CM-2 Baseline Configuration
- CM-6 Configuration Settings
- CM-7 Least Functionality
- SC-7 Boundary Protection
- SI-3 Malicious Code Protection
- SI-4 System Monitoring

Detection

- CAR-2021-05-005: BITSAdmin Download File
- CAR-2021-05-006: CertUtil Download With URLCache and Split Arguments
- CAR-2021-05-007: CertUtil Download With VerifyCtl and Split Arguments
- https://github.com/SigmaHQ/sigma/blob/08ca62cc8860f4660e945805d0dd615ce75258c1/rules/windows/process_creation/win_susp_msiexec_web_install.yml
- https://github.com/SigmaHQ/sigma/blob/08ca62cc8860f4660e945805d0dd615ce75258c1/rules/windows/network_connection/sysmon_win_binary_susp_com.yml
- https://github.com/SigmaHQ/sigma/blob/30bee7204cc1b98a47635ed8e52f44fdf776c602/rules/windows/process_creation/win_susp_wuauclt.yml

Remaining Sub-Techniques Analysis

We opted to look at sub-techniques separately in order to not introduce a down weighting effect on the core techniques. In other words, we wanted to make sure we addressed the overall technique used by adversaries, not just the mechanics of how they accomplished the technique. When we looked solely at the 15 most observed sub-techniques, however, we saw several sub-techniques for which the

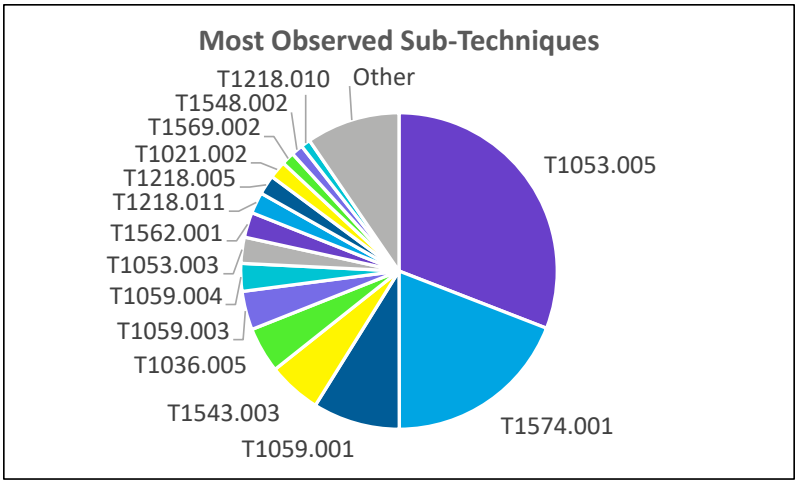


Figure 10 Sub-Techniques Reported in Full Dataset

core technique was not present in the top 15. Defense information for these sub-techniques is provided below.

Service Execution [T1569.002]

There were more than 9,000 observations of *Service Execution* [T1569.002] in our dataset. Adversaries typically can use this technique for one-time or temporary service execution. It can also be used in conjunction with *Windows Service* [T1543.003], one of the top 15 techniques, to achieve persistence or privilege escalation.

Prevention

Thirteen security controls can help mitigate adversary use of *Service Execution* [T1569.002].

- AC-2 Account Management
- AC-3 Access Enforcement
- AC-5 Separation of Duties
- AC-6 Least Privilege
- CA-7 Continuous Monitoring
- CM-2 Baseline Configuration
- CM-5 Access Restrictions for Change
- CM-6 Configuration Settings
- CM-7 Least Functionality
- IA-2 Identification and Authentication (organizational Users)
- SI-3 Malicious Code Protection
- SI-4 System Monitoring
- SI-7 Software, Firmware, and Information Integrity

Detection

- CAR-2013-04-002: Quick execution of a series of suspicious commands
- CAR-2014-02-001: Service Binary Modifications
- CAR-2014-03-005: Remotely Launched Executables via Services
- CAR-2021-05-012: Create Service In Suspicious File Path
- https://github.com/SigmaHQ/sigma/blob/e9260679d4aeae7f696001c5b14d318d31c8f076/rules/windows/driver_load/driver_load_powershell_script_installed_as_service.yml
- https://github.com/SigmaHQ/sigma/blob/71625c54f0469b6c8edad6fb1a3c23dcf17687e5/rules/windows/builtin/win_susp_proceshacker.yml

Bypass User Account Control [T1548.002]

Bypass User Account Control [T1548.002] was observed nearly 9,000 times in our dataset. This technique applies to Windows systems. One common way to *Bypass User Account Control* [T1548.002] leverages *Rundll32* [T1218.011], one of the top techniques addressed above.

Prevention

There are multiple ways to *Bypass User Account Control* [T1548.002], so defense guidance highlights focusing on mitigations. Twelve security controls can help mitigate adversary attempts to *Bypass User Account Control* [T1548.002].

- AC-2 Account Management
- AC-3 Access Enforcement
- AC-5 Separation of Duties
- AC-6 Least Privilege
- CA-8 Penetration Testing
- CM-2 Baseline Configuration
- CM-5 Access Restrictions for Change
- CM-6 Configuration Settings
- IA-2 Identification and Authentication (organizational Users)
- RA-5 Vulnerability Monitoring and Scanning
- SI-2 Flaw Remediation
- SI-4 System Monitoring

Detection

- CAR-2013-10-002: DLL Injection via Load Library
- CAR-2019-04-001: UAC Bypass
- CAR-2021-01-008: Disable UAC
- https://github.com/SigmaHQ/sigma/blob/b731c2059445eef53e37232a5f3634c3473aae0c/rules/windows/process_creation/win_hktl_uacme_uac_bypass.yml
- https://github.com/SigmaHQ/sigma/blob/b731c2059445eef53e37232a5f3634c3473aae0c/rules/windows/file_event/sysmon_uac_bypass_consent_comctl32.yml
- https://github.com/SigmaHQ/sigma/blob/b731c2059445eef53e37232a5f3634c3473aae0c/rules/windows/process_creation/win_uac_bypass_ieinstal.yml

Defenses in Summary

When we look across the preventative controls for the most observed techniques, many controls appear multiple times. The below controls provide the broadest coverage for techniques. Analyzing the relevant controls also gave us greater insight into common adversary behaviors.

1. SI-4 System Monitoring
2. CM-6 Configuration Settings
3. CM-2 Baseline Configuration
4. CM-7 Least Functionality
5. AC-3 Access Enforcement
6. AC-6 Least Privilege
7. AC-2 Account Management
8. AC-5 Separation of Duties
9. CM-5 Access Restrictions for Change
10. IA-2 Identification and Authentication (Organizational Users)

Detecting the adversary begins by monitoring internal systems and external interfaces with a system. Thus, it is no surprise SI-4 System Monitoring is the most associated control. The CAR analytics and Sigma rules provided above support this control.

The next two controls both relate to establishing defined parameters and settings for hardware, software, or firmware in an environment, and then maintaining those specifications and their documentation. This emphasis on baseline (CM-6 and CM-2) is expected because many of the most observed techniques leverage legitimate systems administration tools. The best way to detect adversaries that misuse legitimate technologies is to have strong documentation of the baseline configurations. Detecting deviations from the baseline can signal adversary presence in a network.

The CAR and Sigma rules provided above should be tested against the developed baseline to determine potential false positives. This allows organizations to tune monitoring to their environment and can help prevent alert fatigue.

The rest of the controls can be broadly summarized as making sure people can only do what they are supposed to do based on their roles and responsibilities, and that those restrictions are enforced and managed. This idea forms the basis of what many call zero trust computing. As we look at adversary behavior, we can see that applying this idea creates easier pathways for detecting malicious activity that might otherwise escape notice because it appears legitimate. By strictly limiting what users can do, and verifying they are who they say they are, we can better detect adversaries evading detection by impersonating legitimate users and administrators.

Top 15 Techniques by Year

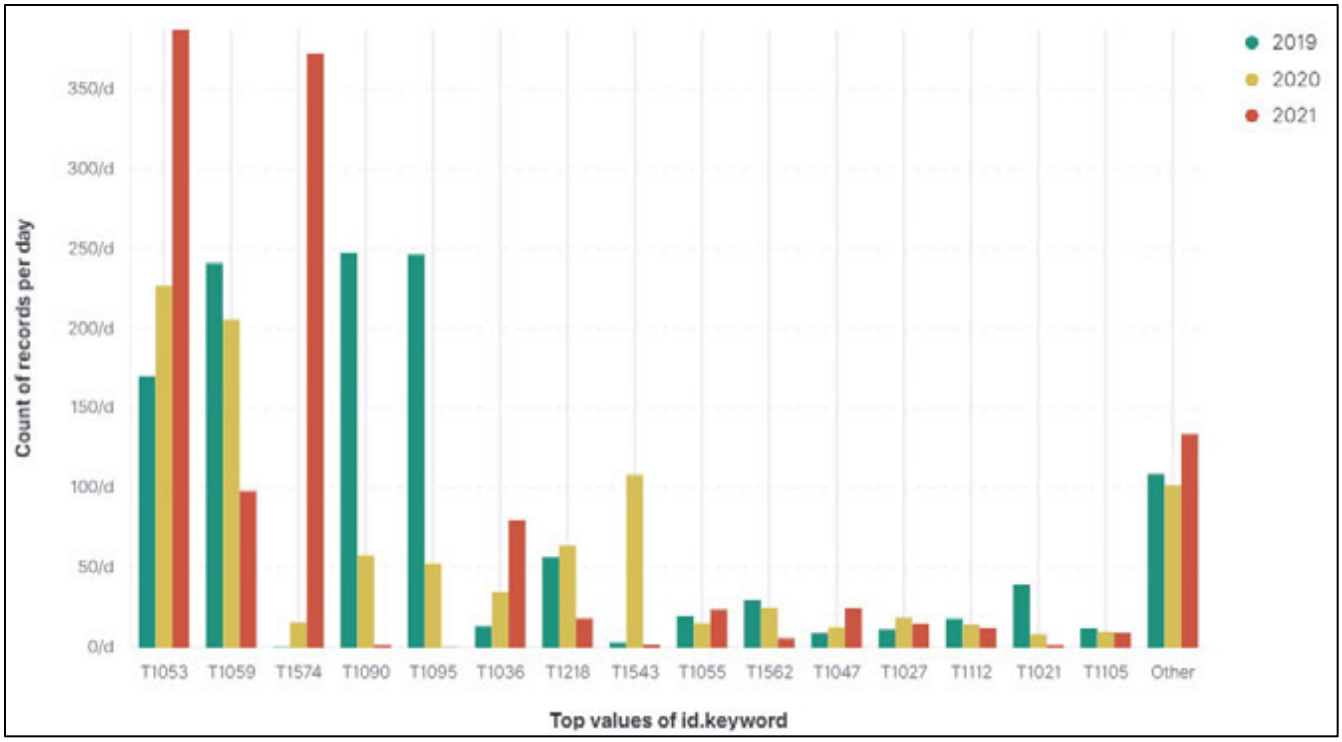


Figure 11 Most Observed Techniques Count Per Day 2019-2021

Understanding how adversaries change their use of techniques provides defenders with additional guidance. Our dataset is skewed towards 2021, so we normalized the data by the techniques observed per day to begin to understand how adversary changed their use of techniques over the years.

Observations of *Scheduled Task/Job* [T1053], *Masquerading* [T1036], *Windows Management Instrumentation* [T1047], and *Hijack Execution Flow* [T1574] increased year over year. *Hijack Execution Flow* [T1574] observations increased from nearly zero in 2019 to the second most prevalent technique observed in 2020. While this might appear anomalous, *Hijack Execution Flow* [T1574] was only added to ATT&CK on 12 March 2020. This technique exposes one weakness of doing time series analysis of adversary techniques as adversary tradecraft and ATT&CK continuously evolve. The other three increases could be indicative of increased adversary use of the techniques, but they might also be reflective of an improved ability to detect these techniques.

Signed Binary/Proxy Execution [T1218], *Process Injection* [T1055], *Obfuscated Files or Information* [T1027], and *Ingress Tool Transfer* [T1105] observations remained relatively consistent year over year. The techniques for which observations have increased or remained relatively stable present the best opportunities to improve defenses.

Observations of *Command and Scripting Interpreter* [T1059], *Remote Services* [T1021], *Impair Defenses* [T1562], and *Modify Registry* [T1112] decreased year over year. With these decreased observations, we can be more confident that it represents decreased adversary use of the technique because it is unlikely

detections will become less effective over time. It does not indicate, however, an overall decrease in adversary activity because adversaries are likely shifting to other techniques with less developed detections.

Proxy [T1090] and *Non-Application Layer Protocol* [T1095] observations both decreased to nearly zero in 2021. We were not able to identify the cause of their decline, as both techniques are likely still used by adversaries. This potentially exposes an issue with our dataset, or a general shift in how these techniques are detected.

The major spike in *Create or Modify System Process* [T1543] observations in 2020 also appears to expose an issue. There are reports of multiple adversaries using this technique in 2021, meaning the decrease in our 2021 data is more intriguing. We were not able to identify the cause in the scope of this work but look forward to investigating in the future.

Technique Sightings Co-Occurrence Analysis

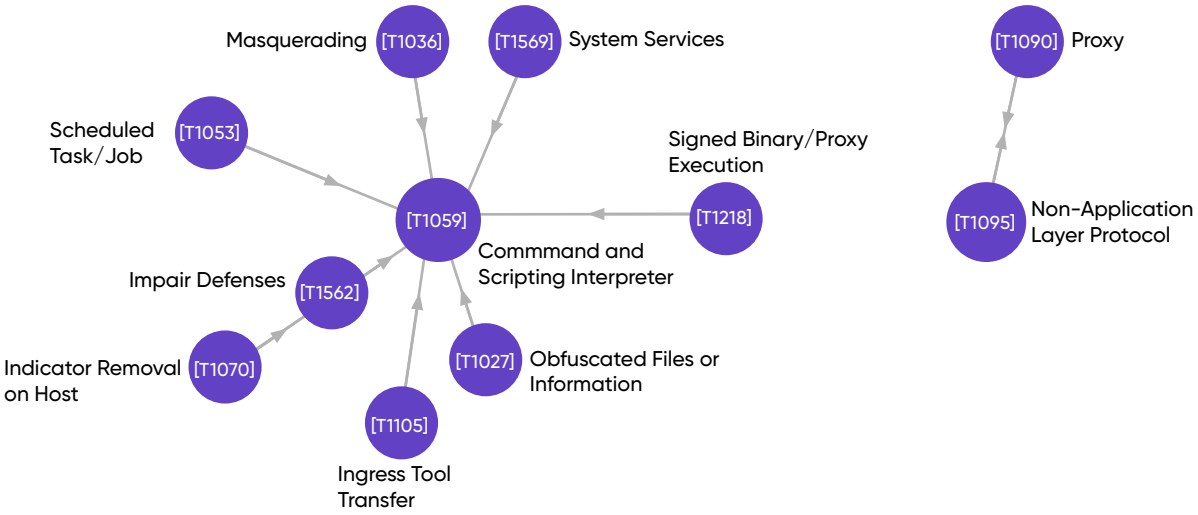


Figure 12 Technique Co-Occurence Graph

One of the initial goals of the Sightings Ecosystem was to identify sequences of TTPs. This is a common question MITRE receives when talking about ATT&CK – given evidence of one technique, what other technique(s) should I look for? [1] We attempted to gather sequence information, but quickly found that the data is often not collected and can be difficult for defenders to ingest in an automated manner.

With our data, we used association rule mining to identify certain TTPs that co-occur, meaning they are frequently observed in the same incident. For this analysis, techniques are visualized as nodes of a network graph. In the graphic below, an arrow points from A to B in cases where technique-B tends to occur when technique-A occurs. The threshold for this was set to 50 percent. What this means is if technique A occurs n times across all Sightings and technique B occurs with A in at least 0.5n Sightings, this relation makes the graph, and an arrow is drawn from A to B.

Of the nine technique co-occurrence links identified, only one is bi-directional. This indicates *Non-Application Layer Protocol* [T1095] and *Proxy* [T1090] frequently occur together. This connection makes sense because adversaries frequently establish a *Proxy* [T1090] using a *Non-Application Layer Protocol* [T1095] to avoid detection.

Among the unidirectional arrows, we saw one chain of techniques. If *Indicator Removal on Host* [T1070] occurs, *Impair Defenses* [T1562] occurs at least 50 percent of the time. Furthermore, when *Impair Defenses* [T1562] occurs, *Command and Scripting Interpreter* [T1059] is also observed at least 50 percent of the time.

The co-occurrence graph also identified *Command and Scripting Interpreter* [T1059] occurs alongside multiple other techniques at least 50 percent of the time. These techniques include *Scheduled Task/Job* [T1053], *Masquerading* [T1036], *System Services* [T1569], *Signed Binary/Proxy Execution* [T1218], *Obfuscated Files or Information* [T1027], and *Ingress Tool Transfer* [T1105]. These connections all make sense because *Command and Scripting Interpreter* [T1059] is a method adversaries may use to accomplish many other techniques.

Defenders can use this co-occurrence information to inform detections. For example, knowing *Command and Scripting Interpreter* [T1059] occurs alongside *Ingress Tool Transfer* [T1105] at least 50 percent of the time, if a defender sees *Ingress Tool Transfer* [T1105], they can look for those command line operations commonly used to invoke the techniques. When there is a technique with known co-occurrence, defenders can be better prepared to identify additional adversary actions in a network.

Co-occurrence analysis is an emerging area of ATT&CK analysis that begins to address the defenders' need for sequence information. The Center is currently working on a related project called Attack Flow to develop a shareable data format, visualization tools, and examples to represent sequences (flows) of attacks. Stay tuned for more to come on that project. To learn more about how MITRE's CALDERA leverages co-occurrence analysis to emulate adversaries, review the blog post here.

Tactic Sightings

Only 6 out of the 14 tactics in ATT&CK were represented in the 15 most observed techniques. *Defense Evasion* [TA0005] was the most observed tactic. Meanwhile *Reconnaissance* [TA0043], *Resource Development* [TA0042], *Initial Access* [TA0001], *Credential Access* [TA0006], *Discovery* [TA0007], *Collection* [TA0009], *Exfiltration* [TA0010], and *Impact* [TA0040] tactics were not represented at all by the 15 most observed techniques.

The total number of tactics is greater than the number of techniques because some techniques are associated with multiple tactics, making the total number of observed tactics 17. Contributors did not associate any technique Sightings with a tactic, so we do not have information on *how or for what purpose* an adversary used a technique.

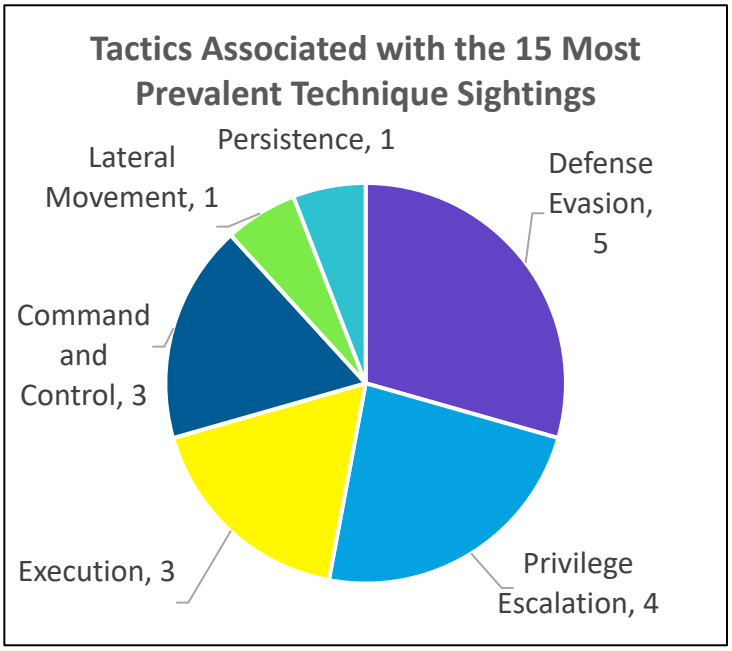


Figure 13 Tactic Breakdown of 15 Most Observed Techniques

When we broaden the tactic analysis to the top 50 techniques, only 3 tactics remained unrepresented: *Reconnaissance* [TA0043], *Resource Development* [TA0042], and *Exfiltration* [TA0010]. *Reconnaissance* [TA0043] and *Resource Development* [TA0042] are likely not represented because there are fewer detections for this type of activity. The majority of *Reconnaissance* [TA0043] and *Resource Development* [TA0042] activity occurs outside of target organization's environment and therefore prior to most contributors' sensors and visibility.

We were surprised, however, to see only one *Lateral Movement* [TA0008] technique in the top 50 techniques. The *Lateral Movement* [TA0008] technique, *Remote Services* [T1021], came in 14th overall in the technique rankings. The limited observations of *Lateral Movement* [TA0008] are likely a reflection of our contributors' visibility.

The absence of *Exfiltration* [TA0010] in the top 50 techniques is also likely caused by our contributors' visibility. Our contributors are focused on preventing attacks and do not have sensors in place to detect data exfiltration in many cases. This does not mean the tactic is not used.

Technique 50, *Software Discovery* [T1518], had less than 500 observations. The remaining 63 techniques represent only a fraction of the Sightings dataset, reducing the insights from a more granular analysis. The heatmap shown below shows the 50 techniques which accounted for much of the activity in our dataset. The coloring is based on how much of our dataset for which the technique accounted.

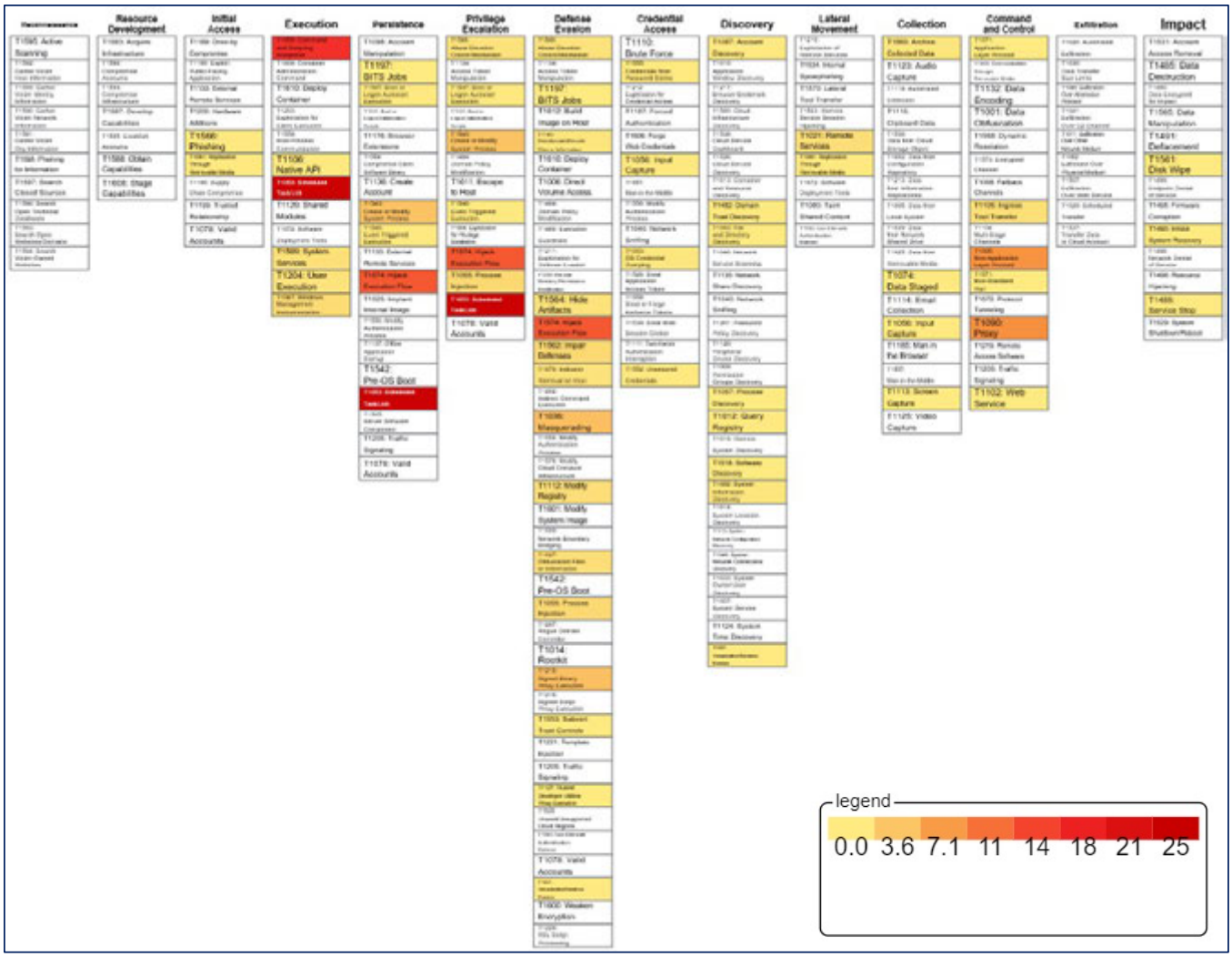


Figure 14 Heatmap of the 50 Most Observed Techniques Scored by the Percentage of the Total Dataset Represented

The Tactic observations are likely a reflection of the ability to detect certain techniques and the location of our contributors' collection efforts. The gaps present an opportunity for the future of the Sightings Ecosystem: will we see improved coverage when we add new contributors? If we do not, some of the gaps likely represent an opportunity for improved detections and sensing technology. It could also mean adversaries are not using those techniques. This still provides value and can provide a data-driven approach to inform the ATT&CK team's decision to deprecate techniques.

If having more contributors resolves gaps, that can inform network defenders about tool acquisition – to gain full coverage of tactics and techniques in ATT&CK, you need multiple tools. The ATT&CK Evaluations project can help inform defenders today on the relative coverage of their security providers.

Detections Lessons Learned

Sightings Ecosystem set out to learn more about adversary behaviors. But, like most analysis, we ended up learning a lot about the defenders as well.

One of early lessons was the important of tuning platforms. Mentioned above, in the complete dataset, one contributor reported a high volume of a single technique. That information distorted our results. The spikes in the frequency graph below shows where multiple mis-labelings occurred. Each colored line in the graph below represents a technique and the y axis represents observations of that technique. Those spikes are indicative of suspicious increases in activity. By looking at a frequency analysis, we could clearly see potential mislabeling or mistuned platforms. We worked with contributors to identify the cause of these issues and remove their impact from our analysis. Resolving issues like these is key to accurate alerting, which enables accurate analysis.

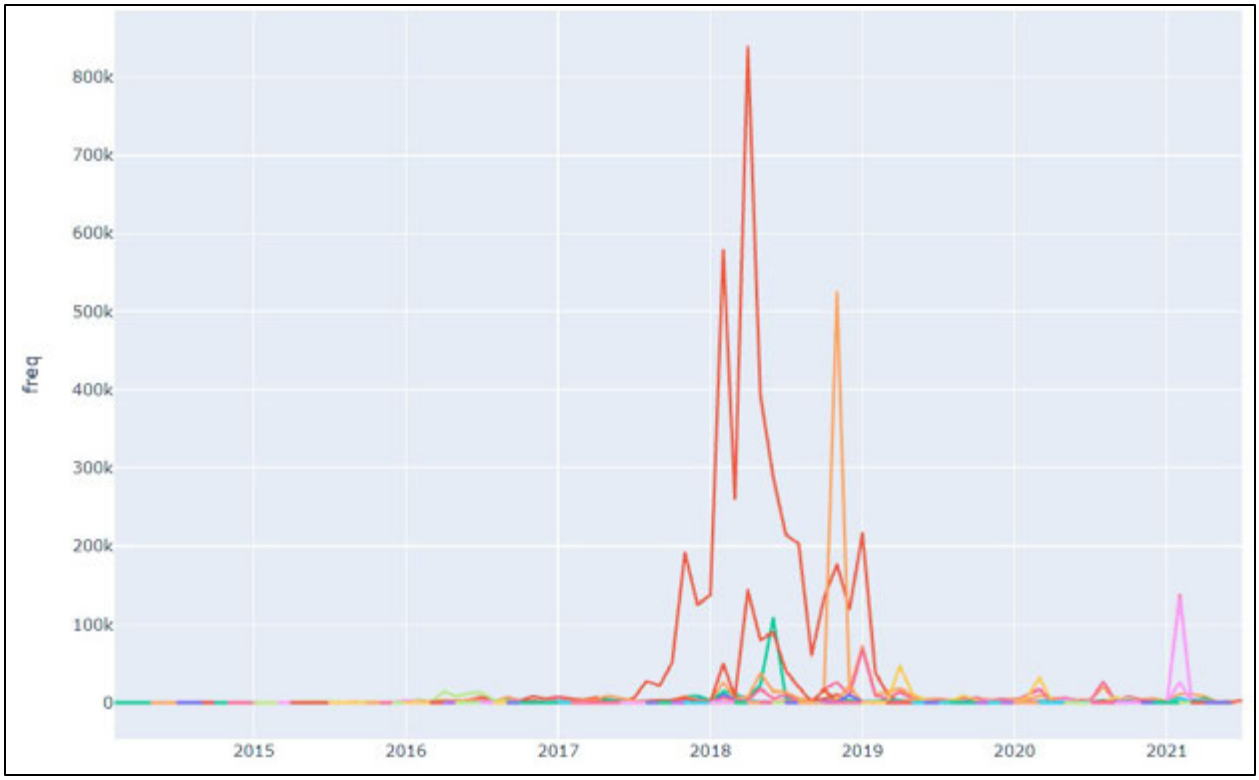


Figure 15 Partial Dataset Technique Frequency Analysis

We ended up selecting a date range based on the mislabeling issues above, but in our final analysis, we identified an additional potential mislabeling. Looking at the frequency analysis of the time-bounded dataset, we can see the potential issues in greater detail. Here again, each colored line represents a single technique, and the y axis reflects the number of observations of the technique.

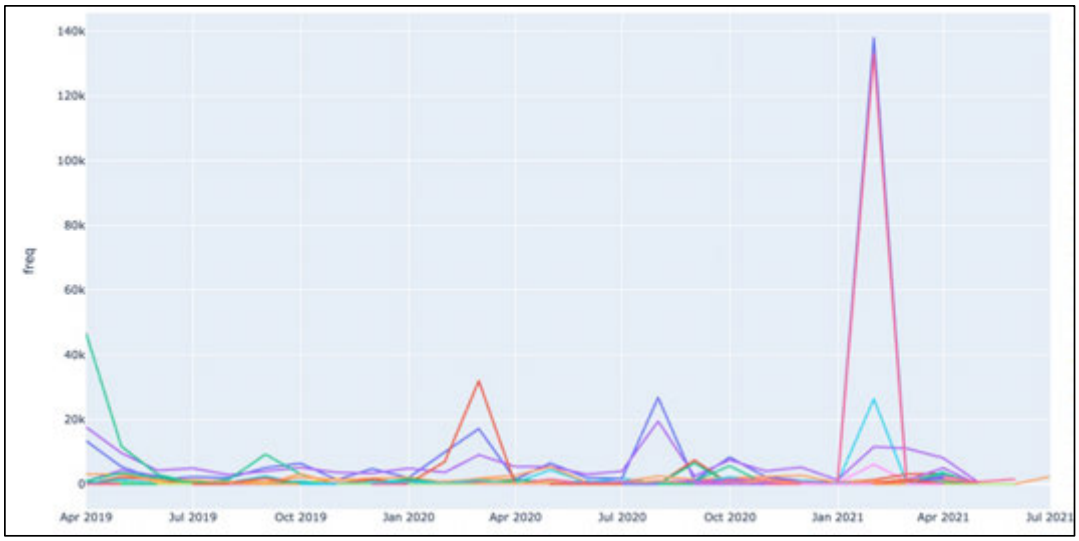


Figure 16 Time-Bounded Dataset Technique Frequency Analysis

In this visualization, there are multiple spikes associated with multiple techniques. These spikes represent potentially anomalous behaviors which can either help us identify issues with our defenses or active adversary campaigns. One key lesson here is the value of frequency analysis. These graphs allow analysts to easily identify potential issues and address them. We were not able to identify the cause of these spikes within the scope of this project, but we will continue to work with contributors to identify the root cause of spikes.

The Sightings Ecosystem accepted raw, human-validated, and machine-validated technique Sightings. We hypothesized that some of the mislabeling issues described above might have resulted from machine-validated data. However, as shown in the graph below, most of the top 15 techniques were human-validated data. Identifying any issues with labeling will require closer collaboration with Sightings contributors.

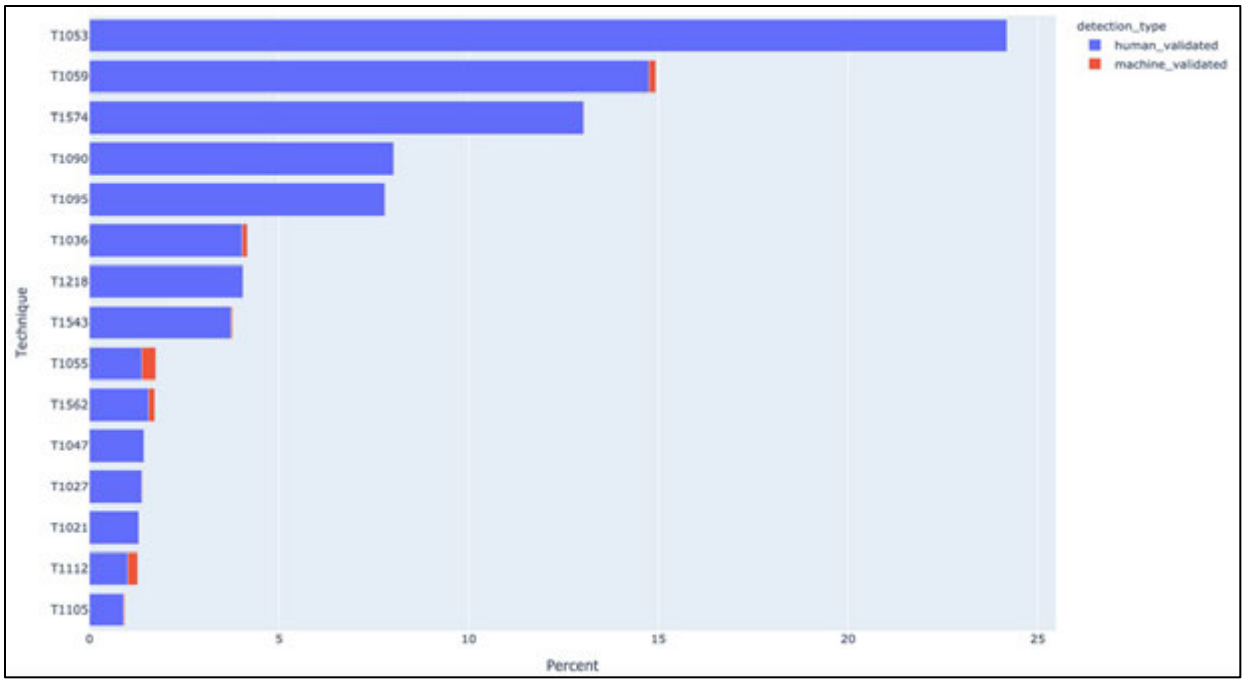


Figure 17 Most Observed Techniques Detection Type Breakdown

Cyber Threat Intelligence Implications Summary

By building an ecosystem of cyber threat actor activities, the Sightings Ecosystem sought to identify and address potential biases in cyber threat intelligence (CTI) reporting. The original intent was to address novelty, visibility, producer, victim, and availability biases. Some we addressed, some we did not.

At the start of the project, we identified several biases common in cyber threat intelligence reporting[2]:

- **Novelty bias:** new and interesting techniques or existing techniques used by new actors get reported, while basic techniques used repeatedly do not.
- **Visibility bias:** organizations publishing intel reports have visibility of certain techniques and not others. Not all techniques can be viewed the same way during or after an incident.
- **Producer bias:** some organizations publish much more reporting, and the types of customers or visibility they have may not reflect the broader industry/world.
- **Victim bias:** certain types of victim organizations may be more likely to report (or be reported on) than others.
- **Availability bias:** techniques that are easily called to mind are likely reported more frequently, as report authors think to include them more often.

We addressed some of these biases, but there were some we were not able to make significant progress towards addressing:

- **Novelty bias:** Success! By looking at all observations of techniques, we overcame the focus on new and interesting techniques. Analysts must justify generating narrative CTI reports because there is not enough time to write up every cyber incident. This can lead to an abundance of reporting focused on new techniques or novel technique sequences. Sightings was able summarize adversary behavior from 800,000 distinct events so novelty was not required to justify a narrative report.
- **Visibility bias:** Our approach did not significantly reduce this bias because we were still limited by the visibility of our contributors. For example, we observed relatively few techniques impacting cloud services or containers, indicating gaps in our dataset. Additionally, we hypothesize that many of the top 15 techniques are those which are easier to detect and label. This means techniques that are harder to observe are not captured. This is a bias we will continue to fight in the CTI and network defense fields.
- **Producer bias:** We need more contributors to make meaningful progress towards this bias. We only had a limited number of contributors and therefore we were not able to build the broadest picture of adversary behavior.
- **Victim bias:** We made some progress towards addressing this bias. By basing our analysis on observed adversary activity, we did not have to rely on victims reporting an incident. However, our visibility is still limited to victims that can afford the solutions offered by our various contributors. Completely

overcoming this bias will be difficult given that threshold. Data-driven CTI analysis will almost always be limited to victims that can afford the sensors to detect and log adversary activity.

- **Availability bias:** By taking a data-driven approach, we significantly reduced this bias in our report. We were not relying on analyst preference for various techniques and were instead only addressing techniques the data provided.

We made progress towards addressing these biases, but this is in no way the complete picture of the cyber threat landscape. The more Sightings Contributors that join the project, the more progress we will make towards addressing these biases. For now, the Sightings Ecosystem is one new window into the cyber threat landscape.

Sightings Data Model and Lessons Learned

By building an ecosystem of cyber threat actor activities, the Sightings Ecosystem sought to identify and address potential biases in cyber threat intelligence (CTI) reporting. The original intent was to address novelty, visibility, producer, victim, and availability biases. Some we addressed, some we did not.

We started with a detailed data model reflecting the objectives of the Sightings Ecosystem project. Our goal was to answer questions about prevalence of techniques by industry, region, company size, user context, and adversary attribution. We were also hoping to understand the most common techniques used by specific malware. These answers would have allowed us to provide more focused insights to defenders. However, we were unable to provide that thorough of an analysis because of a combination of data collection and data analysis issues.

Below is our initial data model. You can read about the data model in-depth here: https://github.com/center-for-threat-informed-defense/sightings_ecosystem/blob/main/data_model.md.

Sighting

Field	Datatype	Description
version	String	A required version string for this model. This must be set to `1.0`.
id	String	A required ID for this event. Can be any format, but if you don't have a preference UUIDv4 is preferred to ensure uniqueness.
sighting_type	String	Either "direct_technique", "direct_software", or "indirect_software". If direct_software or indirect_software, "software_name" field is required.
start_time	Timestamp	The RFC 3339 UTC timestamp of when the activity started.
end_time	Timestamp	The RFC 3339 UTC timestamp of when the activity ended.
detection_type	String	Either "human_validated", "machine_validated", or "raw". Use human_validated when a human analyst has reviewed the detection and determined it to not be a false positive. Use machine_validated when a machine has performed an analysis on it, e.g. a sandbox. Use raw when no validation has occurred.
techniques	List	The list of techniques that were observed. The techniques field has its own schema, so please use the table below for formatting.
hash	String	**Required if using direct_software or indirect_software sighting type.** The MD5, SHA-1, or SHA-256 hash of the software.
software_name	String	The malicious software that was observed. This should ideally be an exact name from the list of Software Names or Associated Software already in ATT&CK.
sector	List[String]	The NAICS code for the sector(s) in which the victim belongs.
country	String	The ISO country code of the victim.
region	String	The IANA Regional Internet Registry code of the victim, e.g. ARIN.
size	String	The approximate size (in number of employees) of the victim. Either "small"(<100), "medium"(100-999), or "large(>999)
attribution_type	String	Either "group", "incident", or "software".
attribution	String	The name of the threat group, incident/campaign, or malicious software associated to this activity. This should ideally be an exact name from the list of Group Names or Associated Groups already in ATT&CK for threat groups, and of Software Names or Associated Software already in ATT&CK for malicious software.
detection_source	String	Either "host_based" or "network_based".
privilege_level	String	Either "system", "admin", "user", or "none".

Techniques

Field	Datatype	Description
technique_id	String	The ATT&CK ID (e.g., "T1086") that was observed.
platform	String	The platform this technique was observed on. Please include the full name, edition, and version. E.g., "Windows 10 Enterprise", "Windows Server 2012 Standard", "MacOS 10.13.5", "Ubuntu 14.04".
start_time	Timestamp	The time that this specific technique was observed.
end_time	Timestamp	The time that this specific technique sighting ended.
tactic	String	The name of the tactic that this technique was used to enable. For example, a sighting of Scheduled Task could indicate whether it was used for privilege escalation or for persistence. Format should be the tactic name as referenced in ATT&CK navigator layer file format (lowercase dashed, credential-access).
raw_data	List[data]	The list of raw data, if sharable, to support this observation. Could be command lines, event records, etc. Format this as described below.

On the data collection front, we were unable to collect information on sector, country, region, size, attribution_type, attribution, or privilege level. We intentionally had an aspirational data model and although we were not successful in gathering this information in this round of the Sightings Ecosystem, we hope future editions will be able to capture some of it.

One of the barriers to collecting the data points above was that our Contributors did not address this information in their existing data collection. Our use case was different than our Contributors' traditional focus, so this gap in data was not unexpected.

Additionally, the structure of the data we received often did not align with our data model. This created some data processing issues and resulted in some back and forth with Contributors. We also eventually had to create some ingest pipelines unique to a Contributor. This causes us to lose some data points along the way. The issue highlighted the importance of using a standard data model, but also the challenges in translating disparate organization's data to that standard data model.

We also encountered issues with some of the data model elements we did receive.

We encountered various hurdles when analyzing the start_time and end_time data elements. Time zone could reveal victim information but requiring Contributors to use UTC removed the opportunity to conduct analysis on the time of day we observe techniques. We were hoping to gain insights into adversaries' hours of operation but were unfortunately limited to more standard time series analysis. Meanwhile, we dropped end_time from the final dataset because no Sightings included an end_time.

The software_name field was also largely unused by contributors. We did receive some file names and some hash values, but we were not set up to do the larger identification of what those hash values represented. In the future, we may need to solicit more context with software and malware Sightings.

Sightings Ecosystem in the Future

ATT&CK is a knowledge base that focuses on reported behavior. However, due to a variety of reasons, the ATT&CK team may lose evidence of specific techniques occurring, leading them to remove those techniques. The Sightings Ecosystem can provide evidence to help the ATT&CK team make a more informed determination. For example, *Shared Webroot* [T1051] was removed because the ATT&CK team did not find evidence of adversary use in the wild. However, this technique was observed multiple times in the Sightings data. This tagging might have been inaccurate, but if it was accurate, sharing those Sightings when they were observed could have led to further discussion on if the technique was in fact defunct.

We also envision the Sightings Ecosystem as an ongoing project that ingests data consistently and whose results are communicated in a dashboard regularly updated with the most observed techniques. This dashboard would allow defenders to view technique prevalence in real time. Through this dashboard, we could understand if headline-grabbing exploits like ProxyLogon are really as widespread as the news suggests, or if some other technique remains more relevant.

There are other opportunities in future Sightings projects. So, what would we do differently if we did Sightings again?

- We accepted human-validated and machine-validated technique Sightings. Luckily, machine-validated technique Sightings made up a relatively small proportion of the data we received. That may change in future iterations. Should we down weight the machine-validated Sightings to reduce the impact of false negatives? Making this call requires answering other questions, including understanding how each contributor defines machine-validated versus human-validated. There are also some concerns about weighting introducing complexity without a similar return on accuracy.

- Going forward, we may need to consider normalizing the date range accepted from contributors. We did not get the same range of data from each contributor. This can influence our understanding of technique frequency. Additionally, if Sightings continue, new Contributors may provide data which changes our previous analysis. While we look forward to having a broader perspective, we need to be prepared for some of the data analysis issues that may arise.

- Could we use honeypots for supplemented data? Many contributors' data systems seek to prevent the threats from taking actions on a victim system or network. Therefore, could we use honeypots to gain a broader understanding of full adversary attack chains?

One question remains: We know adversaries evolve their techniques. We know the sources used by the ATT&CK team cannot cover the full gamut of adversary techniques, nor can it be updated in real-time to include novel adversary techniques. So, how do we detect and conduct time series analysis on techniques not tracked in ATT&CK? Several contributors provided Sightings with IDs unique to their organization. For our purposes, we removed those data points. But, with the release of ATT&CK Workbench, more organizations may be creating techniques unique to their operating space. Can the Sightings Ecosystem be the avenue for transitioning those techniques into ATT&CK? Answering these questions will require a lot more work, and a lot more conversations with MITRE, MITRE Engenuity's Center for Threat-Informed Defense, Center Participants, Contributors, and the larger community.

About the Center for Threat-Informed Defense

The Center is a non-profit, privately funded research and development organization operated by MITRE Engenuity. The Center's mission is to advance the state of the art and the state of the practice in threat-informed defense globally. Comprised of participant organizations from around the globe with highly sophisticated security teams, the Center builds on MITRE ATT&CK®, an important foundation for threat-informed defense used by security teams and vendors in their enterprise security operations. Because the Center operates for the public good, outputs of its research and development are available publicly and for the benefit of all.

https://ctid.mitre-engenuity.org/

For more information:
Center for Threat-Informed Defense
ctid@mitre-engenuity.org