Must Have SOC Analysts customized cookbook

Leverage automation and threat intel data analysis for prioritizing detection

Report customized for telecom sector

This report aims at providing statical analysis of TTPs (Tactics, Techniques and Procedures) used by threat actors targetting telecom sector in order to help SOC in operationalizing their mission.

While contextualising, gathering and analysing available data for a given sector, the overall objective is to introduce a different threat perspective for SOC teams - a perspective based on all known (and shared) threat actor behaviours. The main idea is to provide to SOC team a dedicated baseline to operationalize their efficiency in their daily job from collections to remediations.

The 1st chapter enumerates the threat actors based on MITRE data sources.
The 2nd chapter gives statistics about TTPs and data sources to collect in order to maximise detection capability (beware of bias).
The 3rd and last chapter gives detailed information on how to detect the most used techniques.
This report is AUTOMATICALLY generated based on MITRE ATT&CK and OSSEM data.

MITRE ATT&CK (https://attack.mitre.org) is a globally-accessible knowledge base of adversary tactics and techniques based on real-world observations. The ATT&CK knowledge base is used as a foundation for the development of specific threat models and methodologies in the private sector, in government, and in the cybersecurity product and service community.
With the creation of ATT&CK, MITRE is fulfilling its mission to solve problems for a safer world - by bringing communities together to develop more effective cybersecurity. ATT&CK is open and available to any person or organization for use at no charge.

The OSSEM (Open Source Security Events Metadata / https://github.com/OTRF/OSSEM) is a community-led project that focuses primarily on the documentation and standardization of security event logs from diverse data sources and operating systems. Security events are documented in a dictionary format and can be used as a reference while mapping data sources to data analytics used to validate the detection of adversarial techniques. In addition, the project provides a common data model (CDM) that can be used for data engineers during data normalization procedures to allow security analysts to query and analyze data across diverse data sources. Finally, the project also provides documentation about the structure and relationships identified in specific data sources to facilitate the development of data analytics.

This is a beta version (work still in progress).
Good enough for now.
May this work be of help for you.

Feedbacks, contributions and enrichments are welcome :)
Thomas Billaut <thomas.billaut@protonmail.com>
https://github.com/tbillaut

1. Threat Groups

This chapter aims at giving the list of threat groups targetting the telecom sector.
Data are extracted from MITRE ATT&CK.
Information and citation links can be retrieved from MITRE ATTACK website (https://attack.mitre.org/groups).

1.1 Aquatic Panda

Alias : AQUATIC PANDA

Aquatic Panda (https://attack.mitre.org/groups/G0143) is a suspected China-based threat group with a dual mission of intelligence collection and industrial espionage. Active since at least May 2020, Aquatic Panda (https://attack.mitre.org/groups/G0143) has primarily targeted entities in the telecommunications, technology, and government sectors.
Citation: CrowdStrike AQUATIC PANDA December 2021

1.2 BackdoorDiplomacy

Alias : BackdoorDiplomacy

BackdoorDiplomacy (https://attack.mitre.org/groups/G0135) is a cyber espionage threat group that has been active since at least 2017. BackdoorDiplomacy (https://attack.mitre.org/groups/G0135) has targeted Ministries of Foreign Affairs and telecommunication companies in Africa, Europe, the Middle East, and Asia.
Citation: ESET BackdoorDiplomacy Jun 2021

1.3 APT41

Alias : APT41, WICKED PANDA

APT41 (https://attack.mitre.org/groups/G0096) is a threat group that researchers have assessed as Chinese state-sponsored espionage group that also conducts financially-motivated operations. Active since at least 2012, APT41 (https://attack.mitre.org/groups/G0096) has been observed targeting healthcare, telecom, technology, and video game industries in 14 countries. APT41 (https://attack.mitre.org/groups/G0096) overlaps at least partially with public reporting on groups including BARIUM and Winnti Group (https://attack.mitre.org/groups/G0044).
Citation: FireEye APT41 Aug 2019
Citation: Group IB APT 41 June 2021

1.4 Machete

Alias : Machete, APT-C-43, El Machete

Machete (https://attack.mitre.org/groups/G0095) is a suspected Spanish-speaking cyber espionage group that has been active since at least 2010. It has primarily focused its operations within Latin America, with a particular emphasis on Venezuela, but also in the US, Europe, Russia, and parts of Asia. Machete (https://attack.mitre.org/groups/G0095) generally targets high-profile organizations such as government institutions, intelligence services, and military units, as well as telecommunications and power companies.
Citation: Cylance Machete Mar 2017
Citation: Securelist Machete Aug 2014
Citation: ESET Machete July 2019
Citation: 360 Machete Sep 2020

1.5 GALLIUM

Alias : GALLIUM, Operation Soft Cell

GALLIUM (https://attack.mitre.org/groups/G0093) is a group that has been active since at least 2012, primarily targeting high-profile telecommunications networks. GALLIUM (https://attack.mitre.org/groups/G0093) has been identified in some reporting as likely a Chinese state-sponsored group, based in part on tools used and TTPs commonly associated with Chinese threat actors.
Citation: Cybereason Soft Cell June 2019
Citation: Microsoft GALLIUM December 2019

1.6 APT39

Alias : APT39, REMIX KITTEN, ITG07, Chafer

APT39 (https://attack.mitre.org/groups/G0087) is one of several names for cyberespionage activity conducted by the Iranian Ministry of Intelligence and Security (MOIS) through the front company Rana Intelligence Computing since at least 2014. APT39 (https://attack.mitre.org/groups/G0087) has primarily targeted the travel, hospitality, academic, and telecommunications industries in Iran and across Asia, Africa, Europe, and North America to track individuals and entities considered to be a threat by the MOIS.
Citation: FireEye APT39 Jan 2019
Citation: Symantec Chafer Dec 2015
Citation: FBI FLASH APT39 September 2020
Citation: Dept. of Treasury Iran Sanctions September 2020
Citation: DOJ Iran Indictments September 2020

1.7 APT19

Alias : APT19, Codoso, C0d0so0, Codoso Team, Sunshop Group

APT19 (https://attack.mitre.org/groups/G0073) is a Chinese-based threat group that has targeted a variety of industries, including defense, finance, energy, pharmaceutical, telecommunications, high tech, education, manufacturing, and legal services. In 2017, a phishing campaign was used to target seven law and investment firms.
Citation: FireEye APT19

Some analysts track APT19 (https://attack.mitre.org/groups/G0073) and Deep Panda (https://attack.mitre.org/groups/G0009) as the same group, but it is unclear from open source information if the groups are the same.
Citation: ICIT China's Espionage Jul 2016
Citation: FireEye APT Groups
Citation: Unit 42 C0d0so0 Jan 2016

1.8 Thrip

Alias : Thrip

Thrip (https://attack.mitre.org/groups/G0076) is an espionage group that has targeted satellite communications, telecoms, and defense contractor companies in the U.S. and Southeast Asia. The group uses custom malware as well as "living off the land" techniques.
Citation: Symantec Thrip June 2018

1.9 MuddyWater

Alias : MuddyWater, Earth Vetala , MERCURY, Static Kitten, Seedworm, TEMP.Zagros

MuddyWater (https://attack.mitre.org/groups/G0069) is an Iranian threat group that has primarily targeted Middle Eastern nations, and has also targeted European and North American nations. The group's victims are mainly in the telecommunications, government (IT

services), and oil sectors. Activity from this group was previously linked to FIN7 (https://attack.mitre.org/groups/G0046), but the group is believed to be a distinct group possibly motivated by espionage.
Citation: Unit 42 MuddyWater Nov 2017
Citation: Symantec MuddyWater Dec 2018
Citation: ClearSky MuddyWater Nov 2018
Citation: ClearSky MuddyWater June 2019
Citation: Reaqta MuddyWater November 2017

1.10 OilRig

Alias : OilRig, COBALT GYPSY, IRN2, HELIX KITTEN, APT34

OilRig (https://attack.mitre.org/groups/G0049) is a suspected Iranian threat group that has targeted Middle Eastern and international victims since at least 2014. The group has targeted a variety of sectors, including financial, government, energy, chemical, and telecommunications. It appears the group carries out supply chain attacks, leveraging the trust relationship between organizations to attack their primary targets. FireEye assesses that the group works on behalf of the Iranian government based on infrastructure details that contain references to Iran, use of Iranian infrastructure, and targeting that aligns with nation-state interests.
Citation: Palo Alto OilRig April 2017
Citation: ClearSky OilRig Jan 2017
Citation: Palo Alto OilRig May 2016
Citation: Palo Alto OilRig Oct 2016
Citation: Unit 42 Playbook Dec 2017
Citation: FireEye APT34 Dec 2017
Citation: Unit 42 QUADAGENT July 2018

1.11 APT29

Alias : APT29, IRON RITUAL, IRON HEMLOCK, NobleBaron, Dark Halo, StellarParticle, NOBELIUM, UNC2452, YTTRIUM, The Dukes, Cozy Bear, CozyDuke

APT29 (https://attack.mitre.org/groups/G0016) is threat group that has been attributed to Russia's Foreign Intelligence Service (SVR).
Citation: White House Imposing Costs RU Gov April 2021
Citation: UK Gov Malign RIS Activity April 2021

They have operated since at least 2008, often targeting government networks in Europe and NATO member countries, research institutes, and think tanks. APT29 (https://attack.mitre.org/groups/G0016) reportedly compromised the Democratic National Committee starting in the summer of 2015.
Citation: F-Secure The Dukes
Citation: GRIZZLY STEPPE JAR
Citation: Crowdstrike DNC June 2016
Citation: UK Gov UK Exposes Russia SolarWinds April 2021

In April 2021, the US and UK governments attributed the SolarWinds supply chain compromise cyber operation to the SVR; public statements included citations to APT29 (https://attack.mitre.org/groups/G0016), Cozy Bear, and The Dukes.
Citation: NSA Joint Advisory SVR SolarWinds April 2021
Citation: UK NSCS Russia SolarWinds April 2021

Victims of this campaign included government, consulting, technology, telecom, and other organizations in North America, Europe, Asia, and the Middle East. Industry reporting referred to the actors involved in this campaign as UNC2452, NOBELIUM, StellarParticle, and Dark Halo.
Citation: FireEye SUNBURST Backdoor December 2020

Citation: MSTIC NOBELIUM Mar 2021
Citation: CrowdStrike SUNSPOT Implant January 2021
Citation: Volexity SolarWinds
Citation: Cybersecurity Advisory SVR TTP May 2021

1.12 Deep Panda

Alias : Deep Panda, Shell Crew, WebMasters, KungFu Kittens, PinkPanther, Black Vine

Deep Panda (https://attack.mitre.org/groups/G0009) is a suspected Chinese threat group known to target many industries, including government, defense, financial, and telecommunications.
Citation: Alperovitch 2014

The intrusion into healthcare company Anthem has been attributed to Deep Panda (https://attack.mitre.org/groups/G0009).
Citation: ThreatConnect Anthem

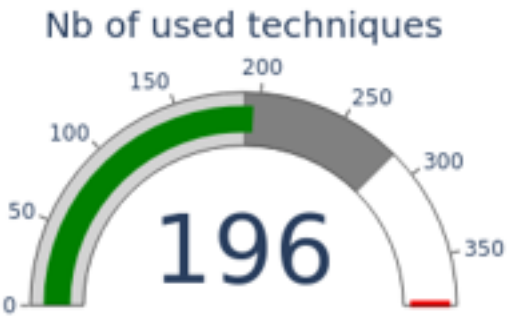
This group is also known as Shell Crew, WebMasters, KungFu Kittens, and PinkPanther.
Citation: RSA Shell Crew

Deep Panda (https://attack.mitre.org/groups/G0009) also appears to be known as Black Vine based on the attribution of both group names to the Anthem intrusion.
Citation: Symantec Black Vine

Some analysts track Deep Panda (https://attack.mitre.org/groups/G0009) and APT19 (https://attack.mitre.org/groups/G0073) as the same group, but it is unclear from open source information if the groups are the same.
Citation: ICIT China's Espionage Jul 2016

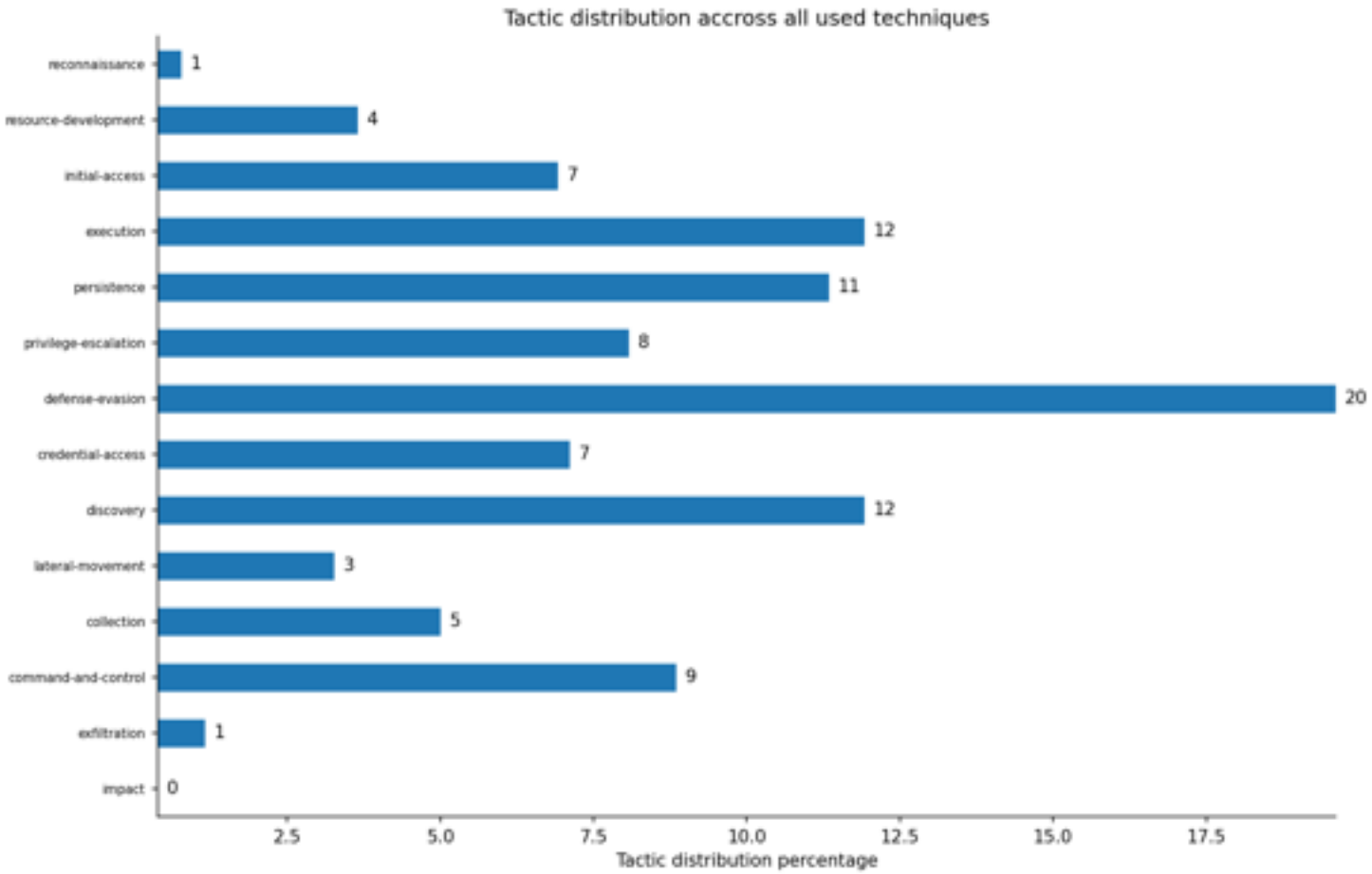
2. What TTPs to prioritize for detection ?

This chapter aims at providing some statistics about tactics and techniques used by the previous threat actors. While understanding most used and share techniques, SOC analysts should be able to focus on most used tactics and techniques. And possibly adopt a new perspective of the priority.



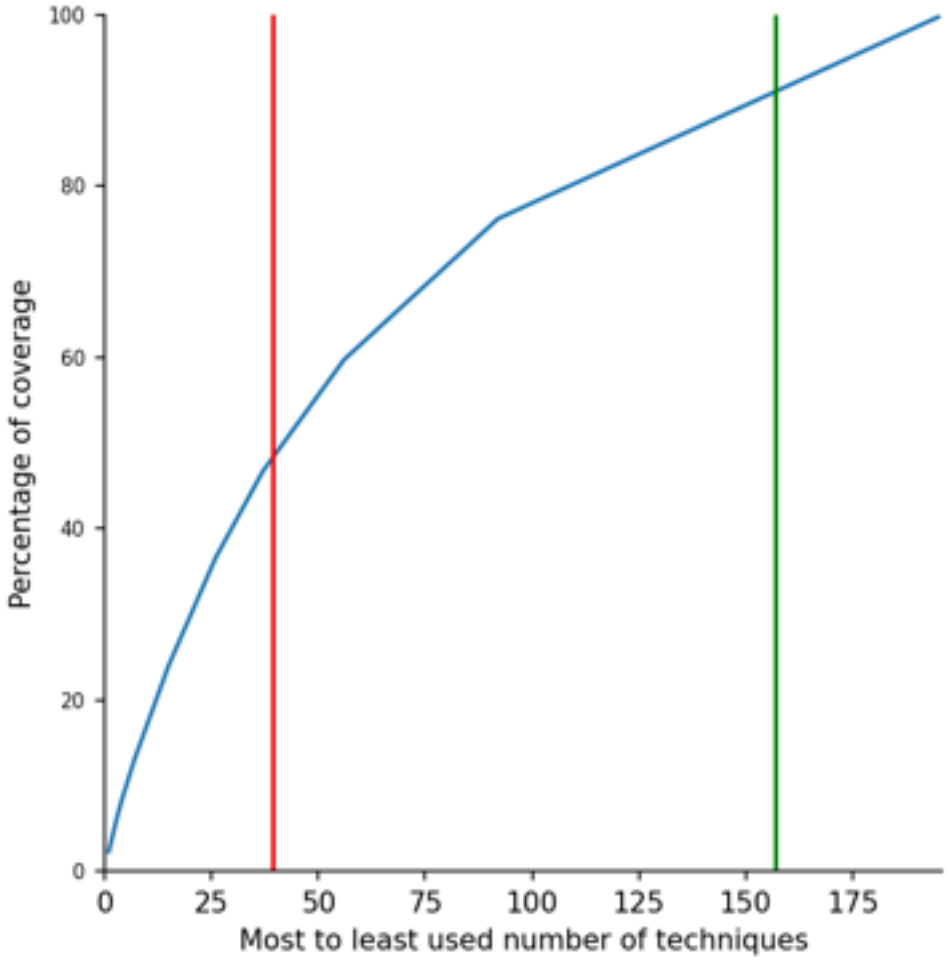
2.1 Tactics distribution

The following chart gives the tactics distribution of all used techniques used by the threat actors.
This representation may offer a new perpective for SOC teams concerning detection capabilities.



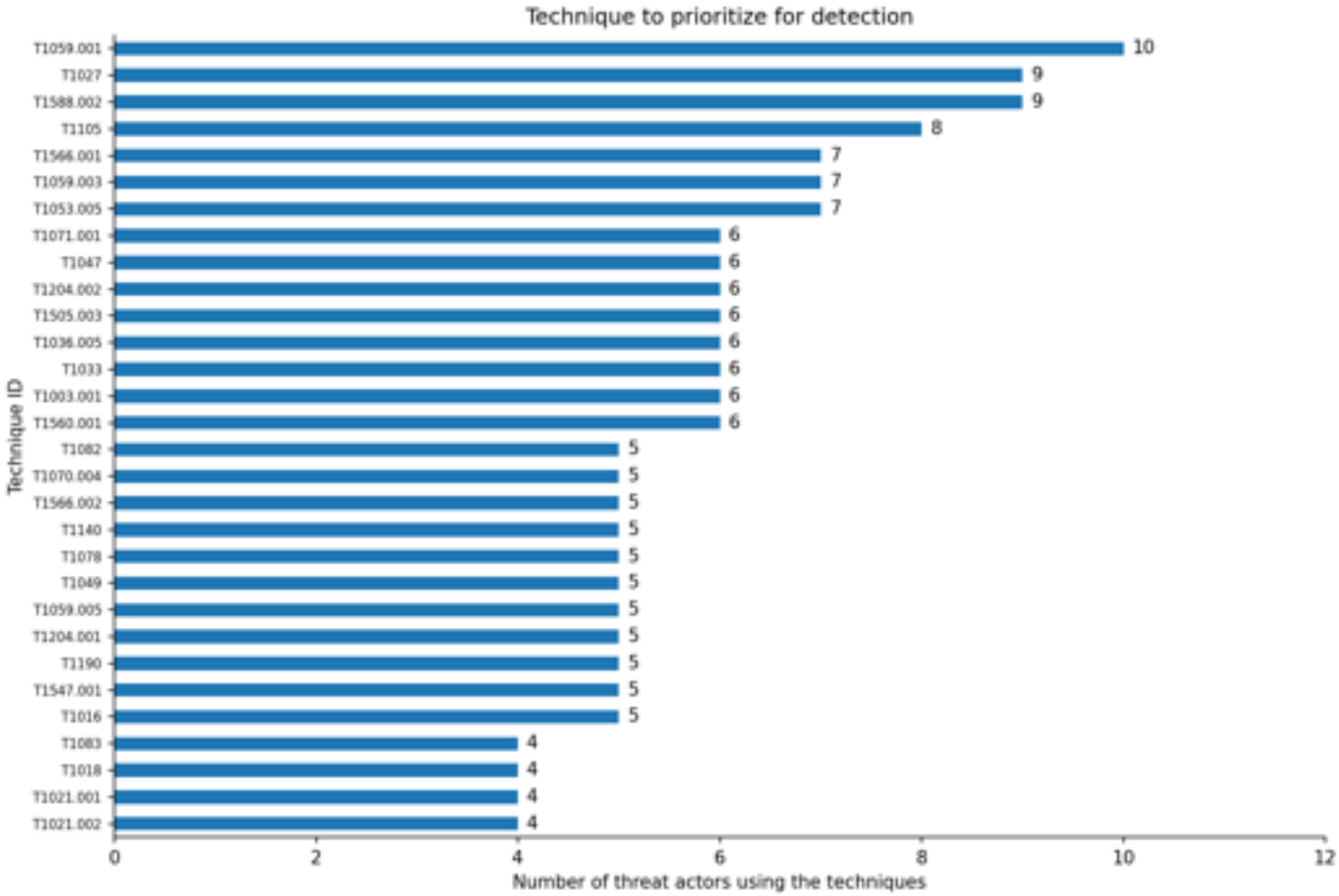
2.2 Technique distribution

The following graph gives the techniques distribution accross all of those threat actors.
It aims at understanding how many techniques need to be covered in order to have the suitable level of detection.
The profile can be compared to the pareto model where covering 20% of the most used techniques would covered 80% of the total of techniques used.
The red line gives the number of techniques corresponding to 20% of total techniques used.
The green line gives the number of techniques corresponding to 80% of total techniques used.



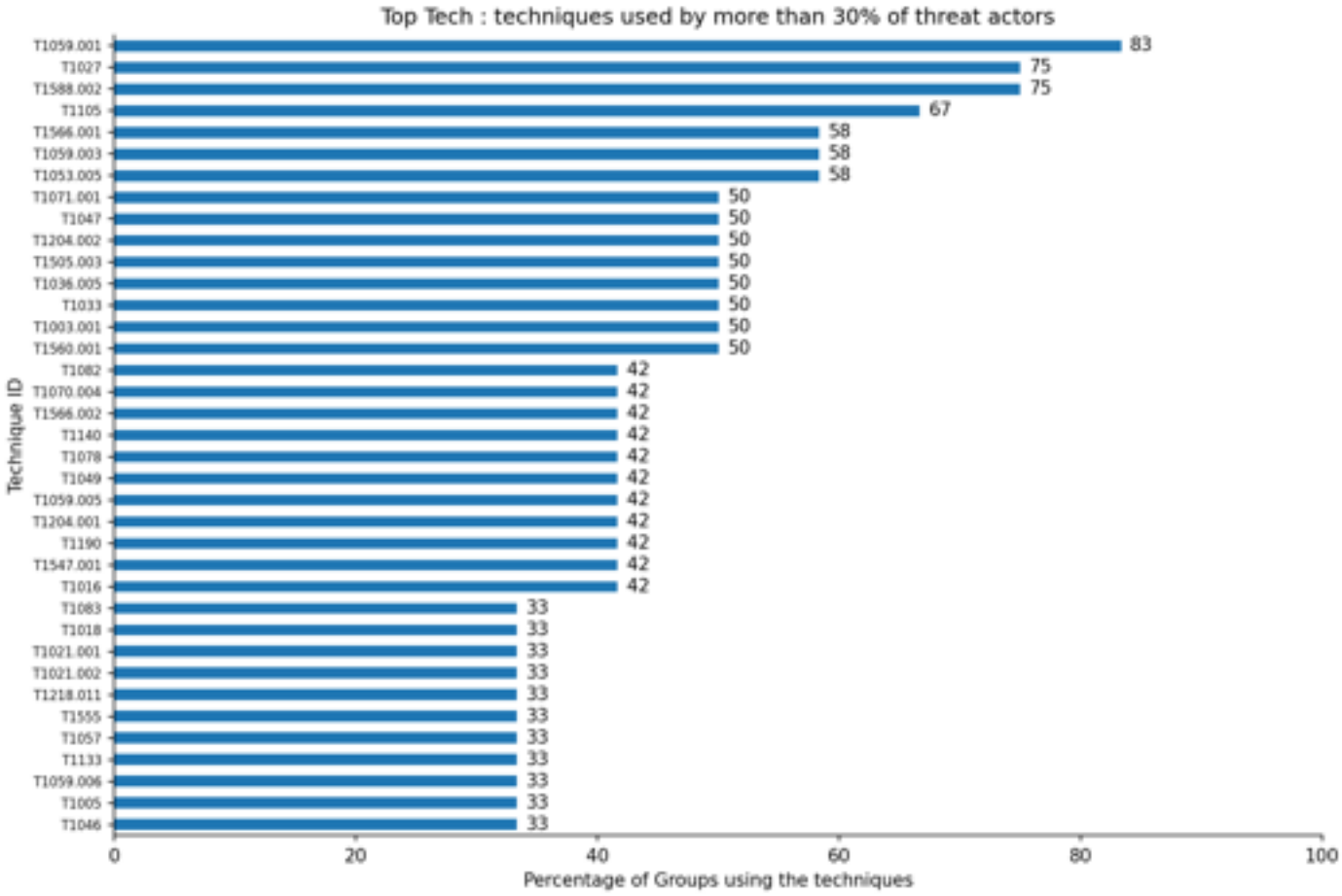
2.3 Top 30 most used techniques

The following graph gives the top 30 techniques that are most used by all of those threat actors.
For each most used technique, the number of group using this technique is given.



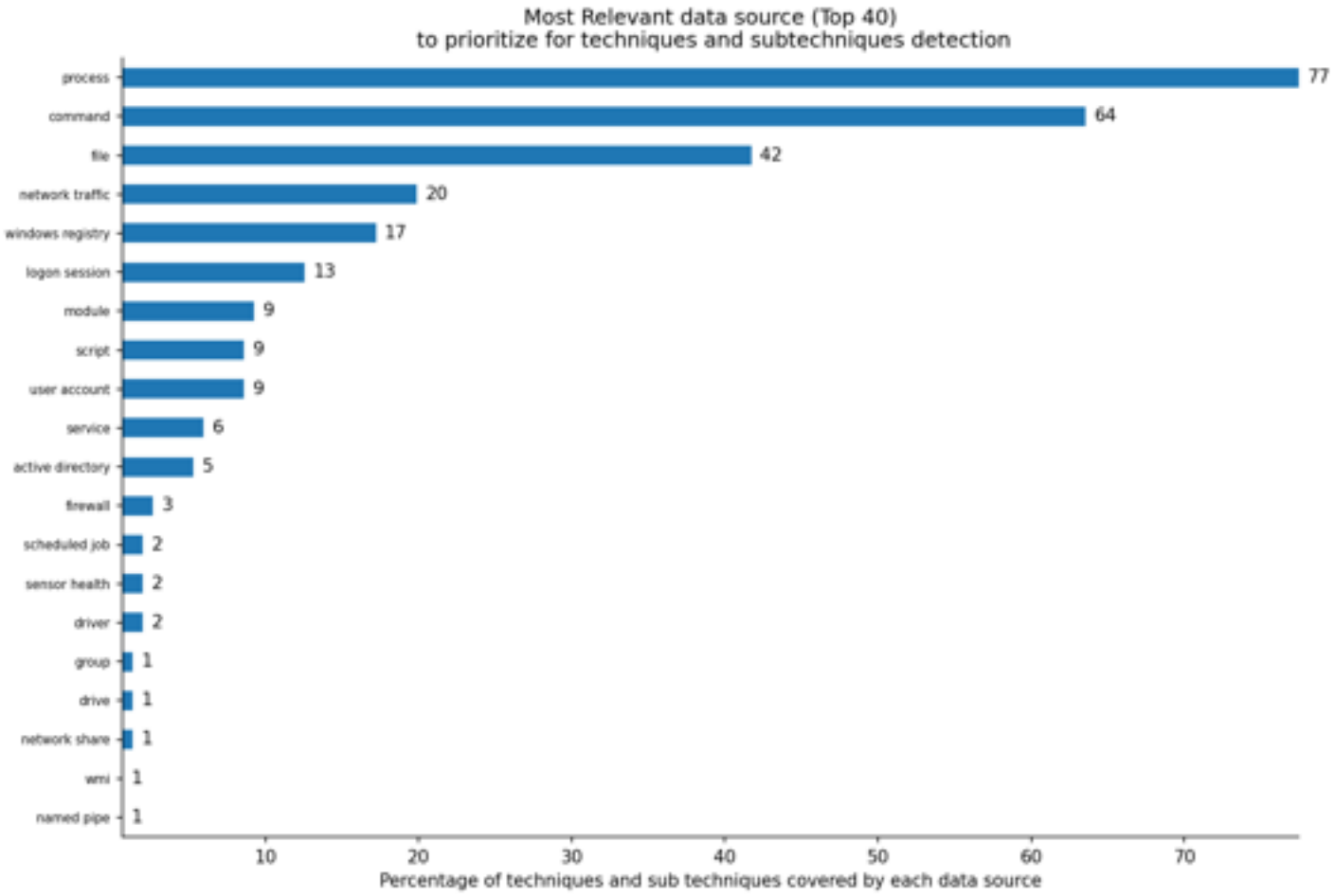
2.4 The Must be covered techniques

The following graph is just a focus of the previous one by giving the techniques that are used by almsot 30% of the threat actors. For each technique, the percentage of threat actors using this technique is given.



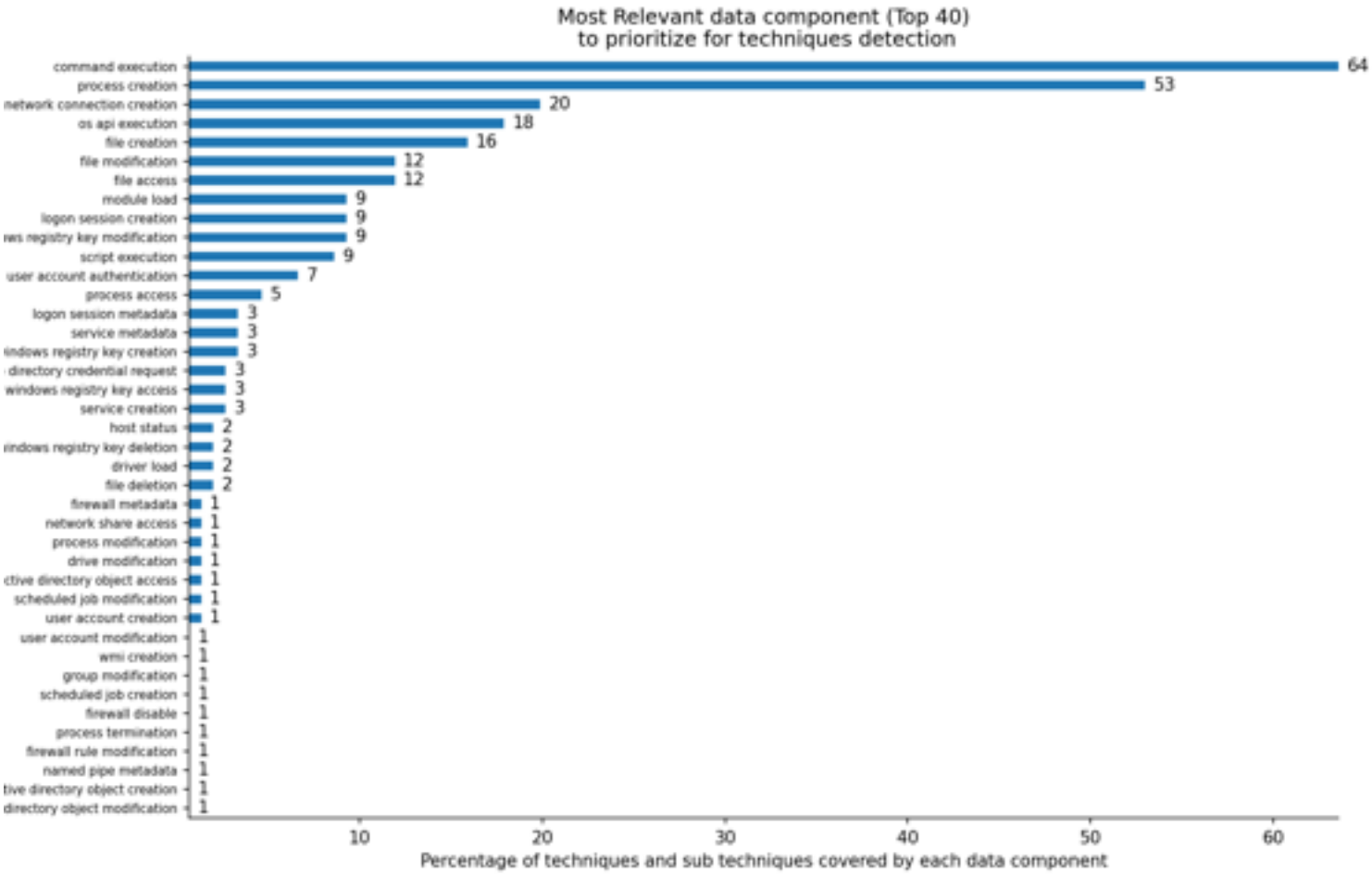
2.5 Top data source to collect for detections

The following graph gives the top 40 data source to collect in order to be able to detect the techniques used by threat actors. Please see annexes for reference.



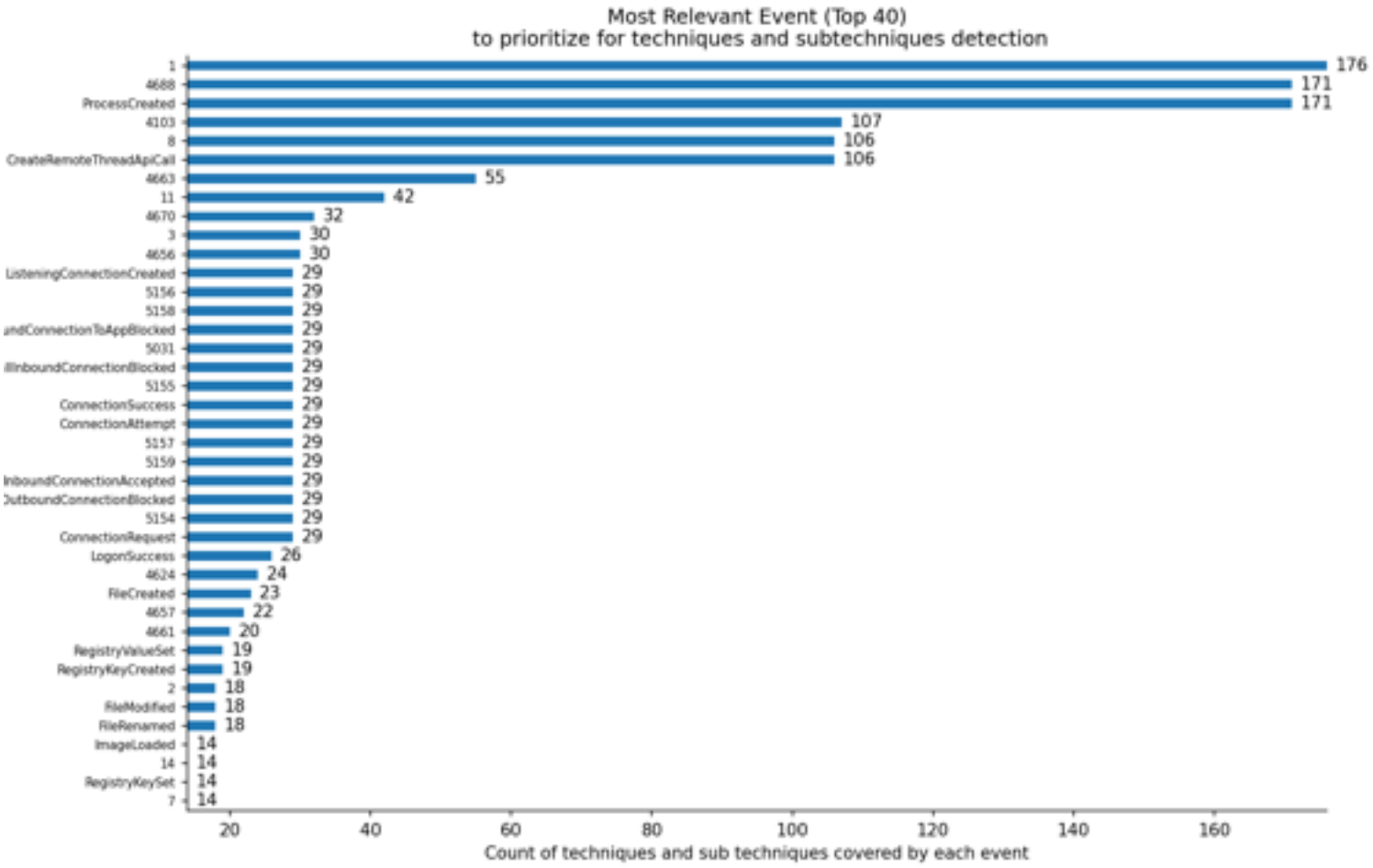
2.6 Top data component to collect for detections

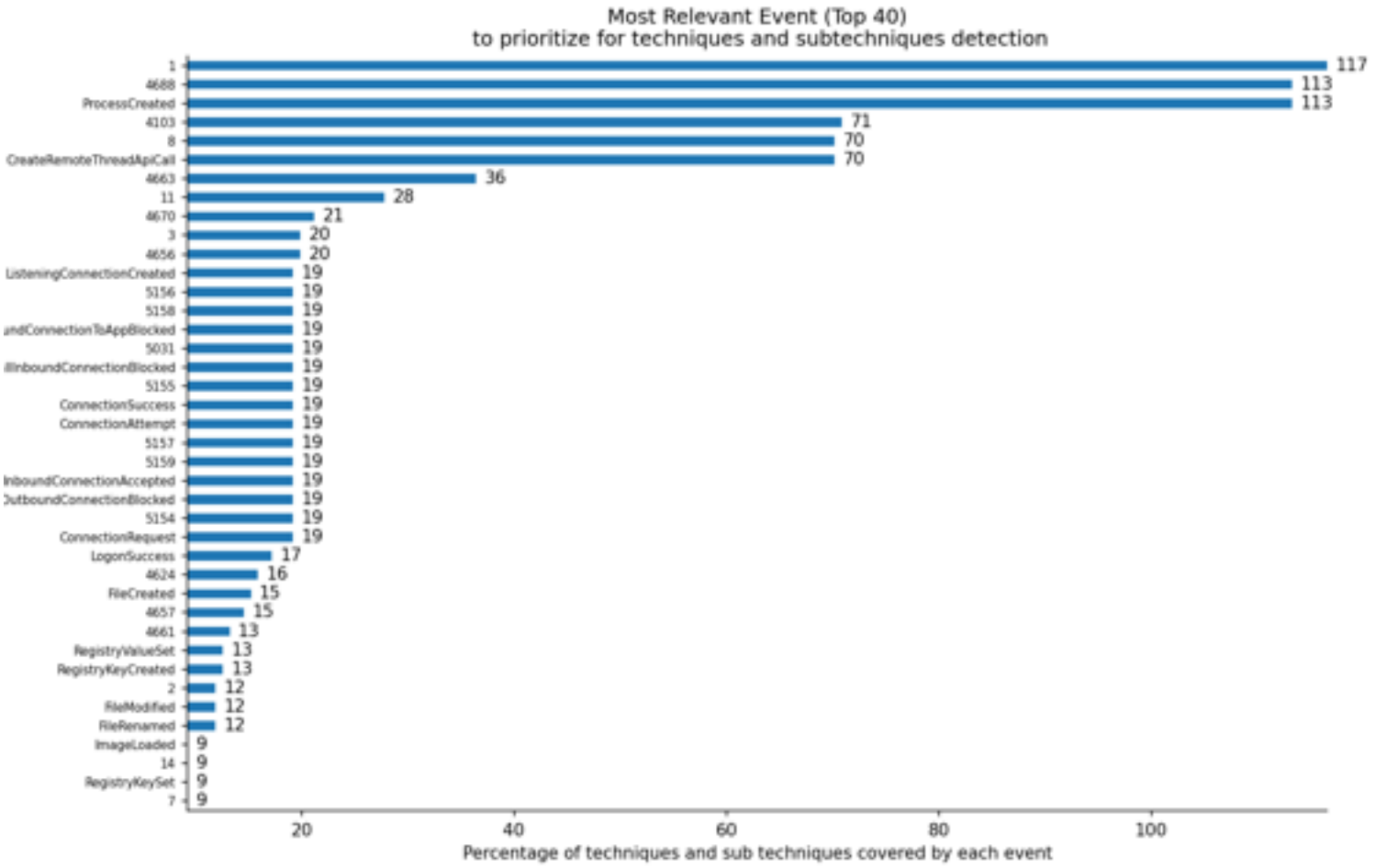
The following graph gives the top 40 data source to collect in order to be able to detect the techniques used by threat actors.
Please see annexes for reference.



2.7 Top event to collect for detections

The following graph gives the top 40 event to collect in order to be able to detect the techniques used by threat actors.
Please see annexes for reference.


Most Relevant Event (Top 40)
to prioritize for techniques and subtechniques detection
1
4688
ProcessCreated
4103
8
CreateRemoteThreadApiCall
4663
11
4670
3
4656
ListeningConnectionCreated
5156
5158
andConnectionToAppBlocked
5031
llInboundConnectionBlocked
5155
ConnectionSuccess
ConnectionAttempt
5157
5159
InboundConnectionAccepted
OutboundConnectionBlocked
5154
ConnectionRequest
LogonSuccess
4624
FileCreated
4657
4661
RegistryValueSet
RegistryKeyCreated
2
FileModified
FileRenamed
ImageLoaded
14
14
RegistryKeySet
7
176
171
171
107
106
106
55
42
32
30
30
29
29
29
29
29
29
29
29
29
29
29
29
29
29
29
26
24
23
22
20
19
19
18
18
18
14
14
14
14
20
40
60
80
100
120
140
160
Count of techniques and sub techniques covered by each event


Most Relevant Event (Top 40)
to prioritize for techniques and subtechniques detection
1
4688
ProcessCreated
4103
8
CreateRemoteThreadApiCall
4663
11
4670
3
4656
ListeningConnectionCreated
5156
5158
andConnectionToAppBlocked
5031
llInboundConnectionBlocked
5155
ConnectionSuccess
ConnectionAttempt
5157
5159
InboundConnectionAccepted
OutboundConnectionBlocked
5154
ConnectionRequest
LogonSuccess
4624
FileCreated
4657
4661
RegistryValueSet
RegistryKeyCreated
2
FileModified
FileRenamed
ImageLoaded
14
RegistryKeySet
7
117
113
113
71
70
70
36
28
21
20
20
19
19
19
19
19
19
19
19
19
19
19
19
19
19
19
17
16
15
15
13
13
13
12
12
12
9
9
9
9
20
40
60
80
100
Percentage of techniques and sub techniques covered by each event

3. How to detect most used techniques ?

This chapter aims at reviewing the most used techniques from most used to least used while providing more detailed information on the technique, the collection data required for detection and how to detect the technique.

3.1 T1059.001

Used by group : Aquatic Panda, APT41, GALLIUM, APT39, APT19, Thrip, MuddyWater, OilRig, APT29, Deep Panda

Tactic : execution

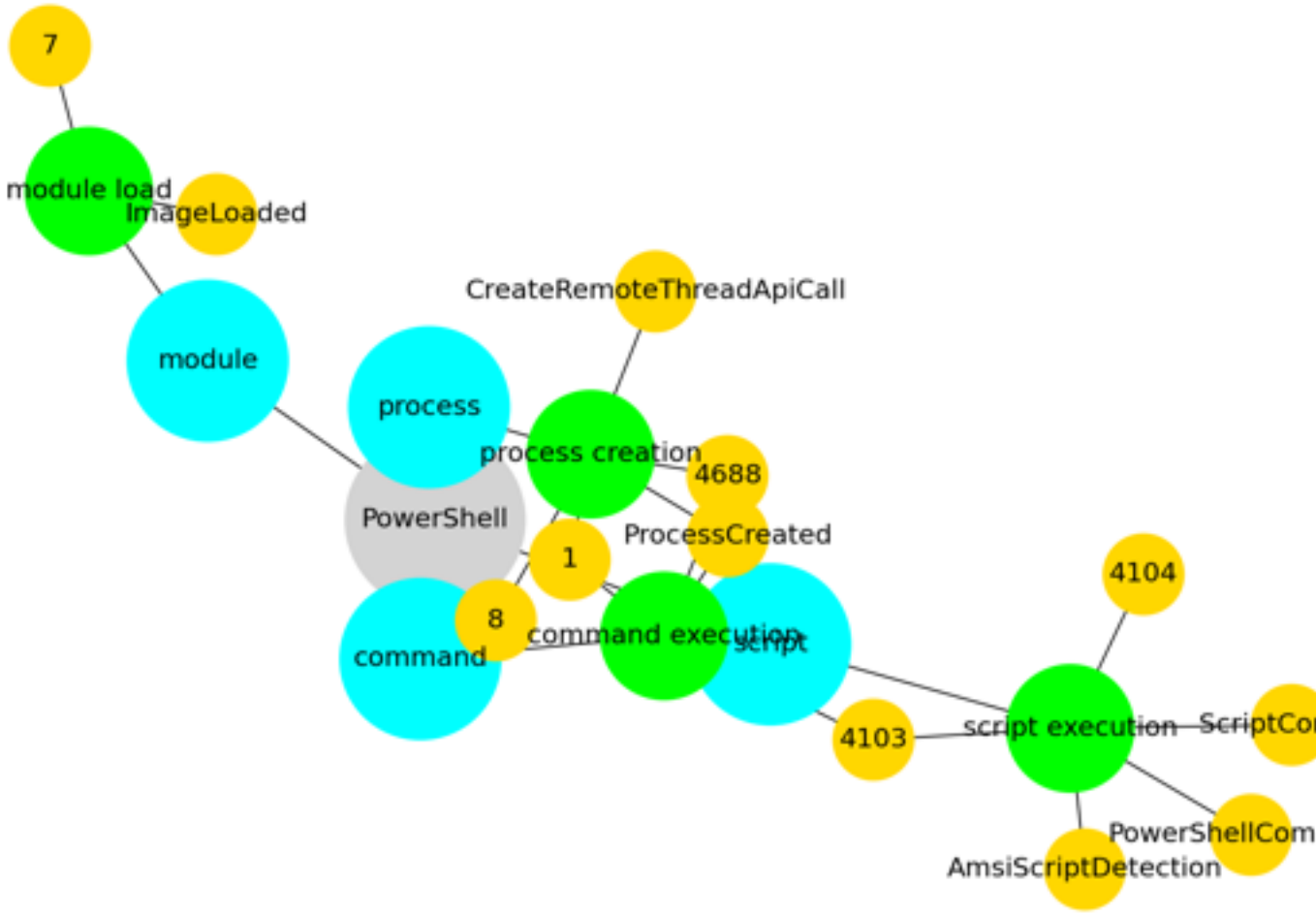
Technique : PowerShell

Adversaries may abuse PowerShell commands and scripts for execution. PowerShell is a powerful interactive command-line interface and scripting environment included in the Windows operating system.(Citation: TechNet PowerShell) Adversaries can use PowerShell to perform a number of actions, including discovery of information and execution of code. Examples include the <code>Start-Process</code> cmdlet which can be used to run an executable and the <code>Invoke-Command</code> cmdlet which runs a command locally or on a remote computer (though administrator permissions are required to use PowerShell to connect to remote systems).

PowerShell may also be used to download and run executables from the Internet, which can be executed from disk or in memory without touching disk.

A number of PowerShell-based offensive testing tools are available, including [Empire](https://attack.mitre.org/software/S0363), [PowerSploit](https://attack.mitre.org/software/S0194), [PoshC2](https://attack.mitre.org/software/S0378), and PSAttack.(Citation: Github PSAttack)

PowerShell commands/scripts can also be executed without directly invoking the <code>powershell.exe</code> binary through interfaces to PowerShell's underlying <code>System.Management.Automation</code> assembly DLL exposed through the .NET framework and Windows Common Language Interface (CLI).(Citation: Sixdub PowerPick Jan 2016)(Citation: SilentBreak Offensive PS Dec 2015)(Citation: Microsoft PSfromCsharp APR 2014)



3.2 T1027

Used by group : Aquatic Panda, BackdoorDiplomacy, APT41, GALLIUM, APT39, APT19, MuddyWater, OilRig, APT29

Tactic : defense-evasion

Technique : Obfuscated Files or Information

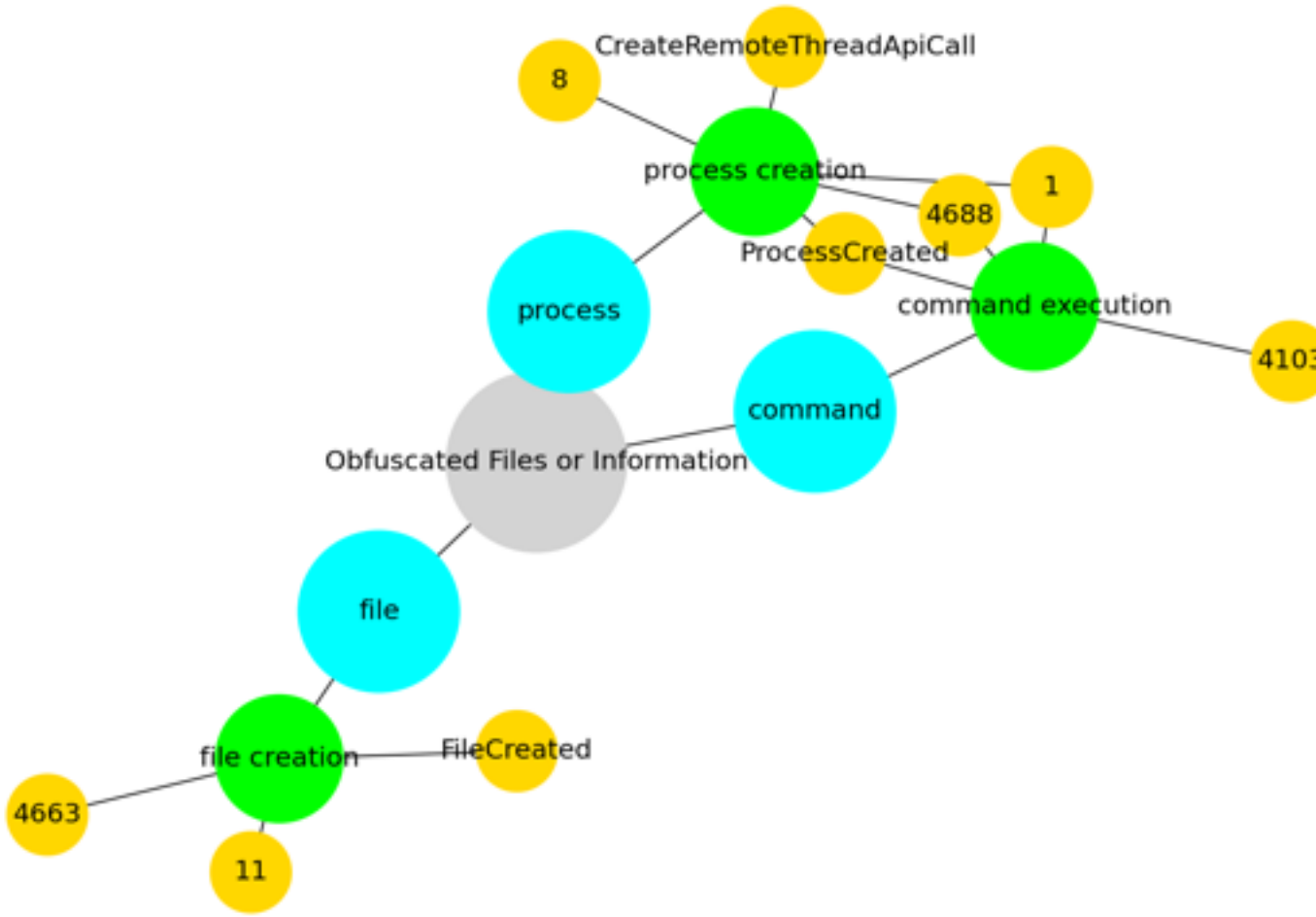
Adversaries may attempt to make an executable or file difficult to discover or analyze by encrypting, encoding, or otherwise obfuscating its contents on the system or in transit. This is common behavior that can be used across different platforms and the network to evade defenses.

Payloads may be compressed, archived, or encrypted in order to avoid detection. These payloads may be used during Initial Access or later to mitigate detection. Sometimes a user's action may be required to open and [Deobfuscate/Decode Files or Information](https://attack.mitre.org/techniques/T1140) for [User Execution](https://attack.mitre.org/techniques/T1204). The user may also be required to input a password to open a password protected compressed/encrypted file that was provided by the adversary.

(Citation: Volexity PowerDuke November 2016) Adversaries may also used compressed or archived scripts, such as JavaScript.

Portions of files can also be encoded to hide the plain-text strings that would otherwise help defenders with discovery. (Citation: Linux/Cdorked.A We Live Security Analysis) Payloads may also be split into separate, seemingly benign files that only reveal malicious functionality when reassembled. (Citation: Carbon Black Obfuscation Sept 2016)

Adversaries may also obfuscate commands executed from payloads or directly via a [Command and Scripting Interpreter](https://attack.mitre.org/techniques/T1059). Environment variables, aliases, characters, and other platform/language specific semantics can be used to evade signature based detections and application control mechanisms. (Citation: FireEye Obfuscation June 2017) (Citation: FireEye Revoke-Obfuscation July 2017)(Citation: PaloAlto EncodedCommand March 2017)



3.3 T1588.002

Used by group : Aquatic Panda, BackdoorDiplomacy, APT41, GALLIUM, APT39, APT19, Thrip, MuddyWater, APT29

Tactic : resource-development

Technique : Tool

Adversaries may buy, steal, or download software tools that can be used during targeting. Tools can be open or closed source, free or commercial. A tool can be used for malicious purposes by an adversary, but (unlike malware) were not intended to be used for those purposes (ex: [PsExec](https://attack.mitre.org/software/S0029)). Tool acquisition can involve the procurement of commercial software licenses, including for red teaming tools such as [Cobalt Strike](https://attack.mitre.org/software/S0154). Commercial software may be obtained through purchase, stealing licenses (or licensed copies of the software), or cracking trial versions.(Citation: Recorded Future Beacon 2019)

Adversaries may obtain tools to support their operations, including to support execution of post-compromise behaviors. In addition to freely downloading or purchasing software, adversaries may steal software and/or software licenses from third-party entities (including other adversaries).

3.4 T1105

Used by group : Aquatic Panda, BackdoorDiplomacy, APT41, GALLIUM, APT39, MuddyWater, OilRig, APT29

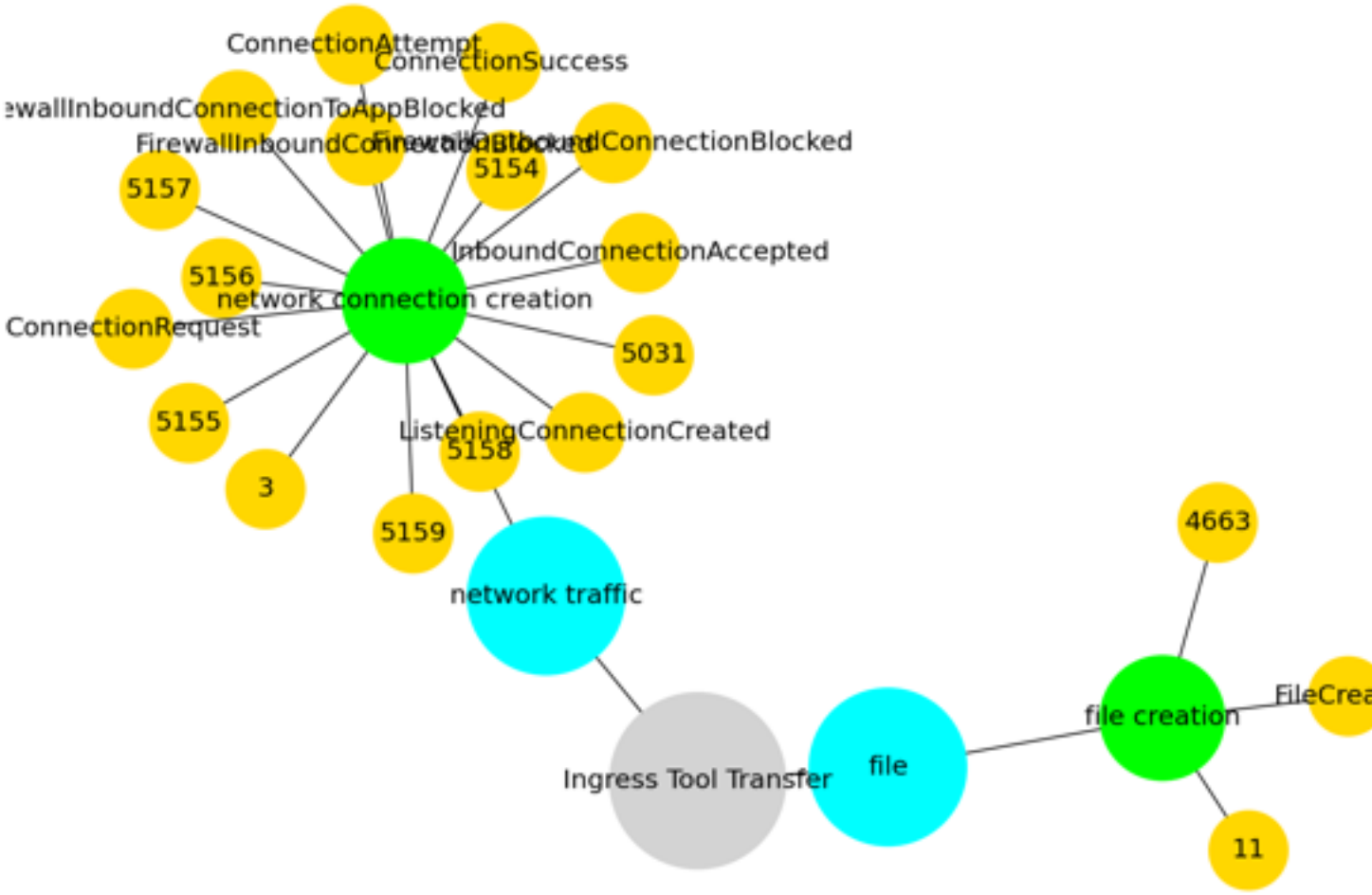
Tactic : command-and-control

Technique : Ingress Tool Transfer

Adversaries may transfer tools or other files from an external system into a compromised environment. Tools or files may be copied from an external adversary-controlled system to the victim network through the command and control channel or through alternate protocols such as [ftp](https://attack.mitre.org/software/S0095). Once present, adversaries may also transfer/spread tools between victim devices within a compromised environment (i.e. [Lateral Tool Transfer](https://attack.mitre.org/techniques/T1570)).

Files can also be transferred using various [Web Service](https://attack.mitre.org/techniques/T1102)s as well as native or otherwise present tools on the victim system.(Citation: PTSecurity Cobalt Dec 2016)

On Windows, adversaries may use various utilities to download tools, such as `copy`, `finger`, and [PowerShell](https://attack.mitre.org/techniques/T1059/001) commands such as <code>IEX(New-Object Net.WebClient).downloadString()</code> and <code>Invoke-WebRequest</code>. On Linux and macOS systems, a variety of utilities also exist, such as `curl`, `scp`, `sftp`, `tftp`, `rsync`, `finger`, and `wget`.(Citation: t1105_lolbas)



3.5 T1566.001

Used by group : APT41, Machete, APT39, APT19, MuddyWater, OilRig, APT29

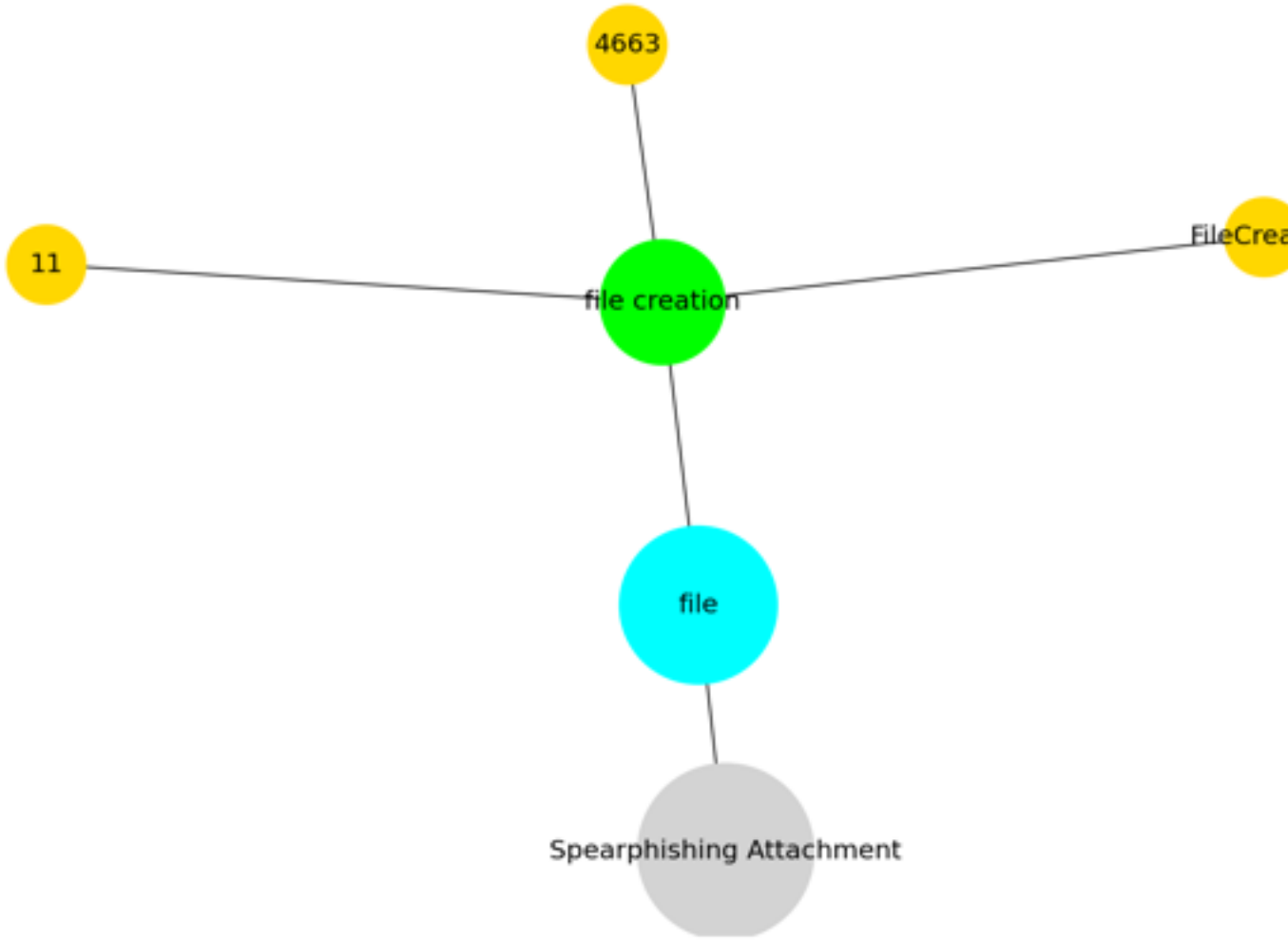
Tactic : initial-access

Technique : Spearphishing Attachment

Adversaries may send spearphishing emails with a malicious attachment in an attempt to gain access to victim systems. Spearphishing attachment is a specific variant of spearphishing. Spearphishing attachment is different from other forms of spearphishing in that it employs the use of malware attached to an email. All forms of spearphishing are electronically delivered social engineering targeted at a specific individual, company, or industry. In this scenario, adversaries attach a file to the spearphishing email and usually rely upon [User Execution](https://attack.mitre.org/techniques/T1204) to gain execution. Spearphishing may also involve social engineering techniques, such as posing as a trusted source.

There are many options for the attachment such as Microsoft Office documents, executables, PDFs, or archived files. Upon opening

the attachment (and potentially clicking past protections), the adversary's payload exploits a vulnerability or directly executes on the user's system. The text of the spearphishing email usually tries to give a plausible reason why the file should be opened, and may explain how to bypass system protections in order to do so. The email may also contain instructions on how to decrypt an attachment, such as a zip file password, in order to evade email boundary defenses. Adversaries frequently manipulate file extensions and icons in order to make attached executables appear to be document files, or files exploiting one application appear to be a file for a different one.



3.6 T1059.003

Used by group : Aquatic Panda, APT41, Machete, GALLIUM, MuddyWater, OilRig, APT29

Tactic : execution

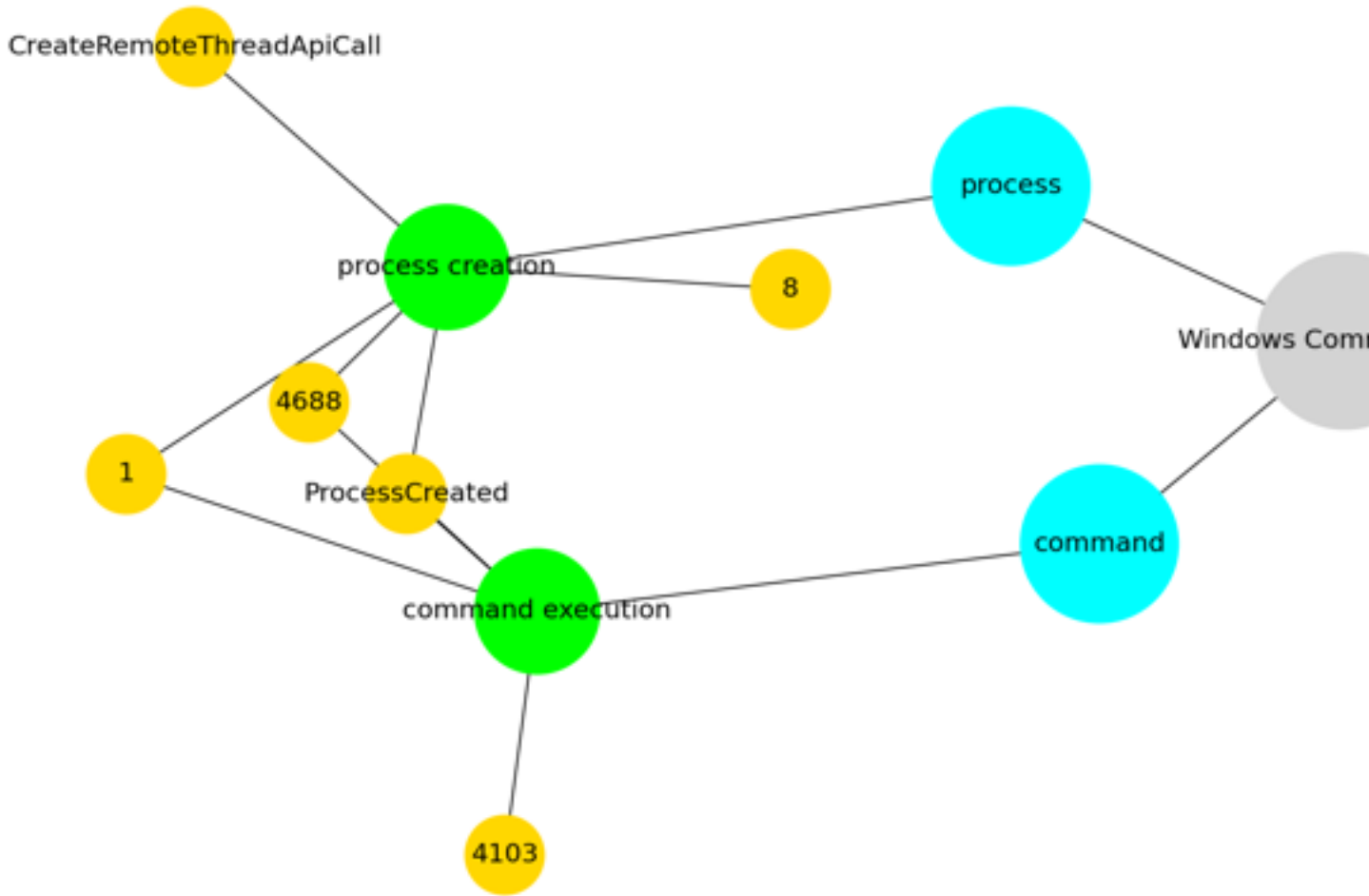
Technique : Windows Command Shell

Adversaries may abuse the Windows command shell for execution. The Windows command shell

([cmd](https://attack.mitre.org/software/S0106)) is the primary command prompt on Windows systems. The Windows command prompt can be used to control almost any aspect of a system, with various permission levels required for different subsets of commands. The command prompt can be invoked remotely via [Remote Services](https://attack.mitre.org/techniques/T1021) such as [SSH](https://attack.mitre.org/techniques/T1021/004).(Citation: SSH in Windows)

Batch files (ex: .bat or .cmd) also provide the shell with a list of sequential commands to run, as well as normal scripting operations such as conditionals and loops. Common uses of batch files include long or repetitive tasks, or the need to run the same set of commands on multiple systems.

Adversaries may leverage [cmd](https://attack.mitre.org/software/S0106) to execute various commands and payloads. Common uses include [cmd](https://attack.mitre.org/software/S0106) to execute a single command, or abusing [cmd](https://attack.mitre.org/software/S0106) interactively with input and output forwarded over a command and control channel.



3.7 T1053.005

Used by group : APT41, Machete, GALLIUM, APT39, MuddyWater, OilRig, APT29

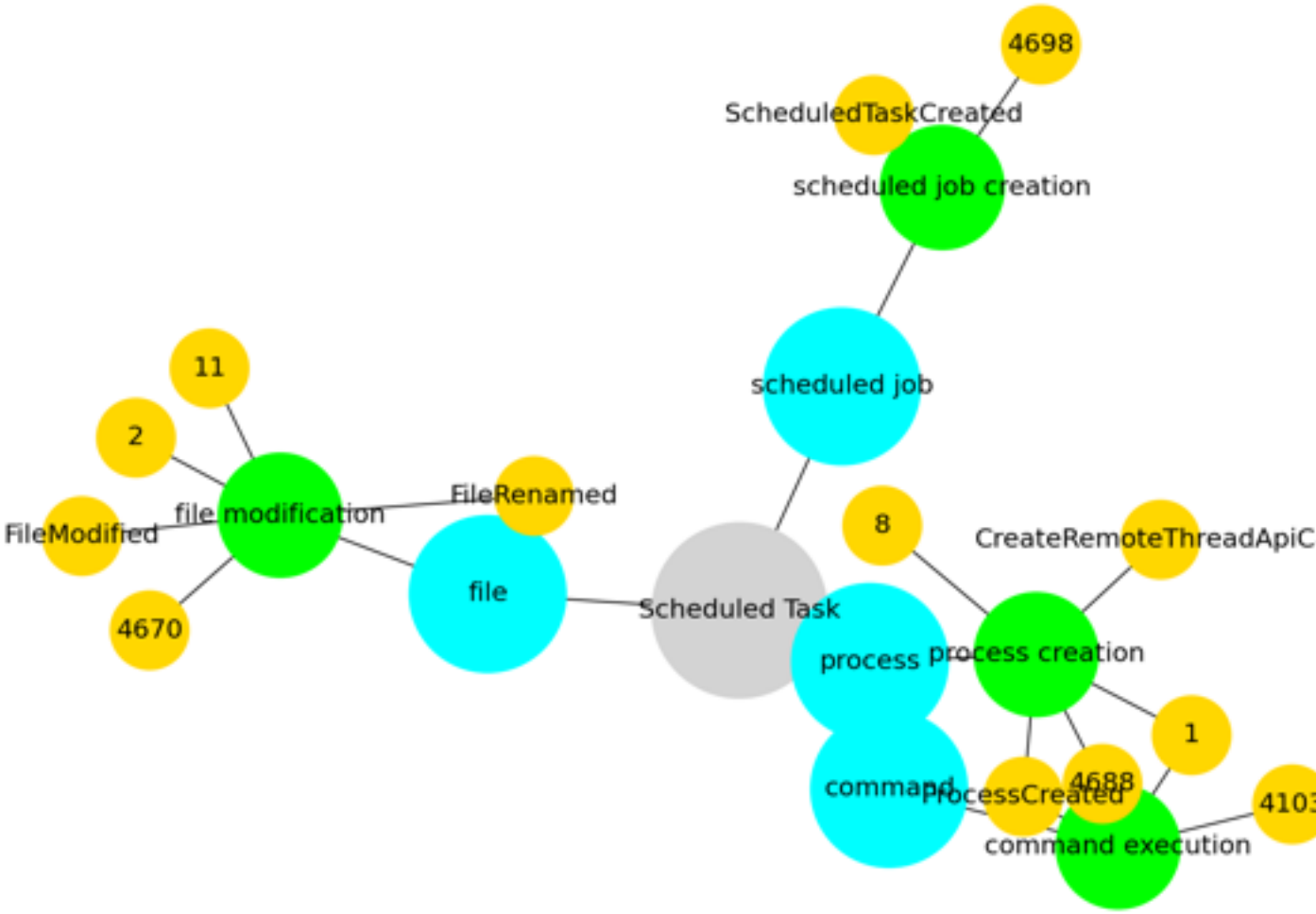
Tactic : execution, persistence, privilege-escalation

Technique : Scheduled Task

Adversaries may abuse the Windows Task Scheduler to perform task scheduling for initial or recurring execution of malicious code. There are multiple ways to access the Task Scheduler in Windows. The [schtasks](https://attack.mitre.org/software/S0111) utility can be run directly on the command line, or the Task Scheduler can be opened through the GUI within the Administrator Tools section of the Control Panel. In some cases, adversaries have used a .NET wrapper for the Windows Task Scheduler, and alternatively, adversaries have used the Windows netapi32 library to create a scheduled task.

The deprecated [at](https://attack.mitre.org/software/S0110) utility could also be abused by adversaries (ex: [At](https://attack.mitre.org/techniques/T1053/002)), though <code>at.exe</code> can not access tasks created with <code>schtasks</code> or the Control Panel.

An adversary may use Windows Task Scheduler to execute programs at system startup or on a scheduled basis for persistence. The Windows Task Scheduler can also be abused to conduct remote Execution as part of Lateral Movement and/or to run a process under the context of a specified account (such as SYSTEM). Similar to [System Binary Proxy Execution](https://attack.mitre.org/techniques/T1218), adversaries have also abused the Windows Task Scheduler to potentially mask one-time execution under signed/trusted system processes.(Citation: ProofPoint Serpent)



3.8 T1071.001

Used by group : APT41, APT39, APT19, MuddyWater, OilRig, APT29

Tactic : command-and-control

Technique : Web Protocols

Adversaries may communicate using application layer protocols associated with web traffic to avoid detection/network filtering by blending in with existing traffic. Commands to the remote system, and often the results of those commands, will be embedded within the protocol traffic between the client and server.

Protocols such as HTTP and HTTPS that carry web traffic may be very common in environments. HTTP/S packets have many fields and headers in which data can be concealed. An adversary may abuse these protocols to communicate with systems under their control within a victim network while also mimicking normal, expected traffic.

3.9 T1047

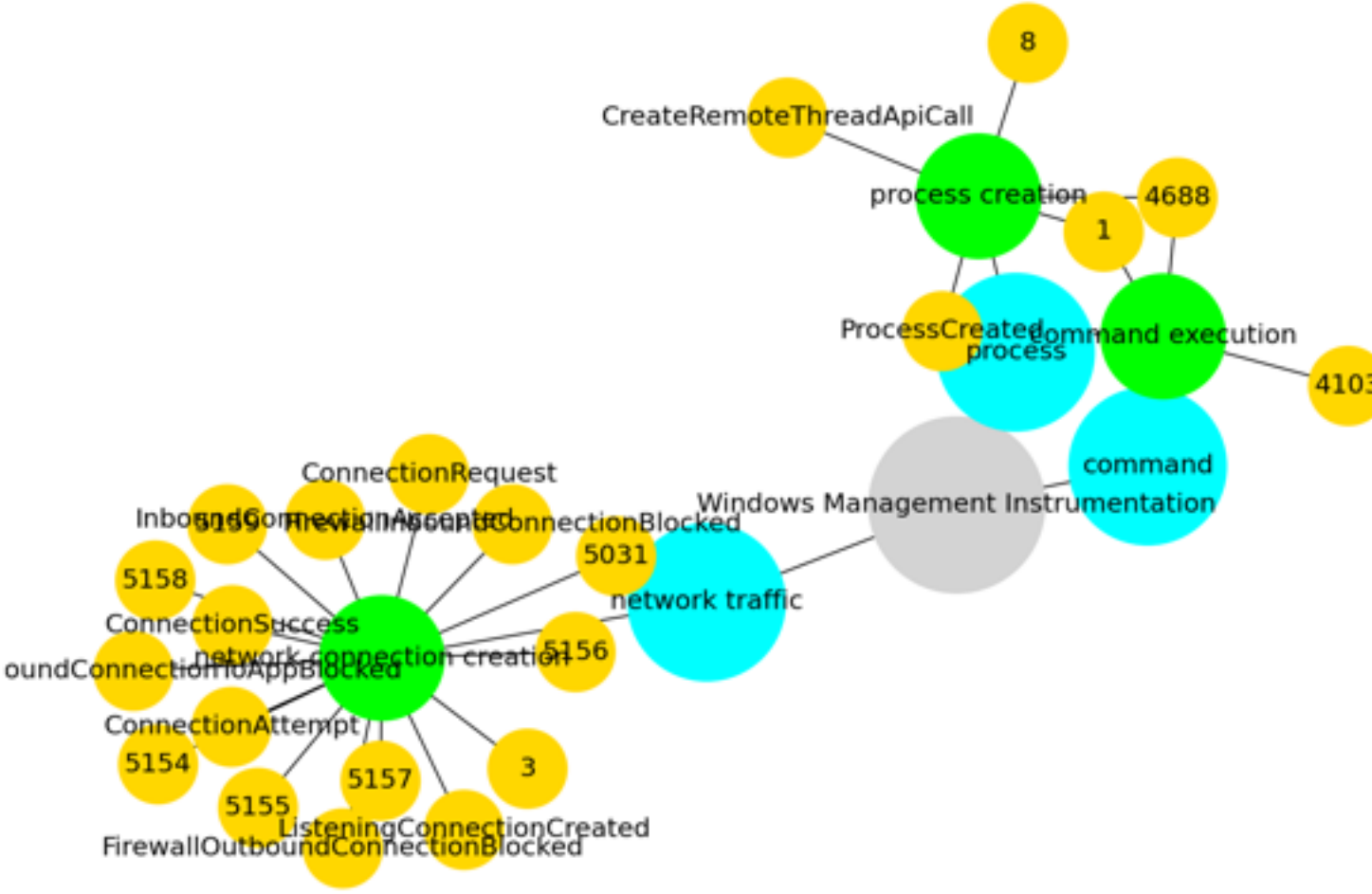
Used by group : APT41, GALLIUM, MuddyWater, OilRig, APT29, Deep Panda

Tactic : execution

Technique : Windows Management Instrumentation

Adversaries may abuse Windows Management Instrumentation (WMI) to execute malicious commands and payloads. WMI is an administration feature that provides a uniform environment to access Windows system components. The WMI service enables both local and remote access, though the latter is facilitated by [Remote Services](https://attack.mitre.org/techniques/T1021) such as [Distributed Component Object Model](https://attack.mitre.org/techniques/T1021/003) (DCOM) and [Windows Remote Management](https://attack.mitre.org/techniques/T1021/006) (WinRM).(Citation: MSDN WMI) Remote WMI over DCOM operates using port 135, whereas WMI over WinRM operates over port 5985 when using HTTP and 5986 for HTTPS.(Citation: MSDN WMI)(Citation: FireEye WMI 2015)

An adversary can use WMI to interact with local and remote systems and use it as a means to execute various behaviors, such as gathering information for Discovery as well as remote Execution of files as part of Lateral Movement. (Citation: FireEye WMI SANS 2015) (Citation: FireEye WMI 2015)



3.10 T1204.002

Used by group : Machete, APT39, APT19, MuddyWater, OilRig, APT29

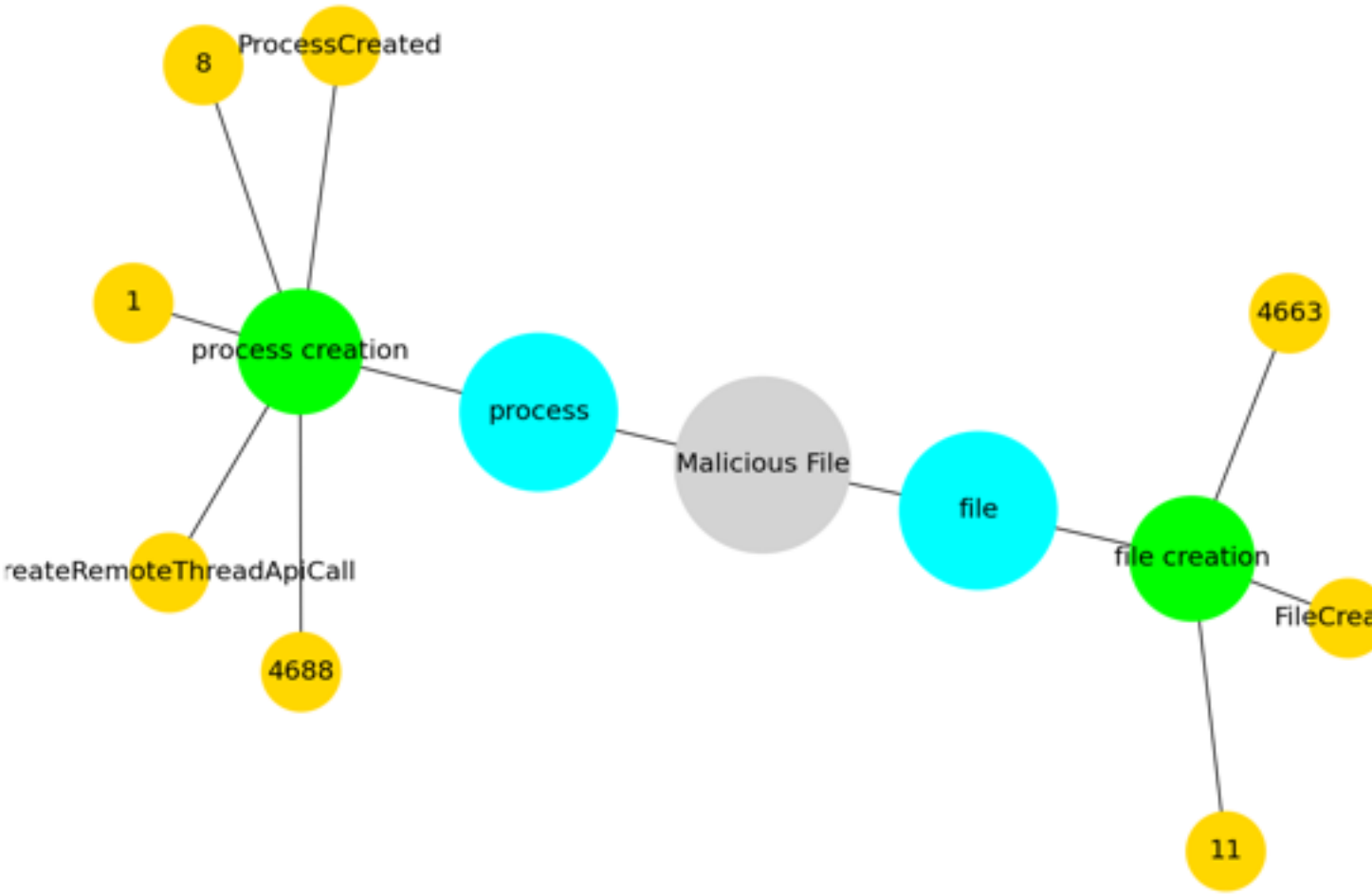
Tactic : execution

Technique : Malicious File

An adversary may rely upon a user opening a malicious file in order to gain execution. Users may be subjected to social engineering to get them to open a file that will lead to code execution. This user action will typically be observed as follow-on behavior from [Spearphishing Attachment](https://attack.mitre.org/techniques/T1566/001). Adversaries may use several types of files that require a user to execute them, including .doc, .pdf, .xls, .rtf, .scr, .exe, .lnk, .pif, and .cpl.

Adversaries may employ various forms of [Masquerading](https://attack.mitre.org/techniques/T1036) and [Obfuscated Files or Information](https://attack.mitre.org/techniques/T1027) to increase the likelihood that a user will open and successfully execute a malicious file. These methods may include using a familiar naming convention and/or password protecting the file and supplying instructions to a user on how to open it.(Citation: Password Protected Word Docs)

While [Malicious File](https://attack.mitre.org/techniques/T1204/002) frequently occurs shortly after Initial Access it may occur at other phases of an intrusion, such as when an adversary places a file in a shared directory or on a user's desktop hoping that a user will click on it. This activity may also be seen shortly after [Internal Spearphishing](https://attack.mitre.org/techniques/T1534).



3.11 T1505.003

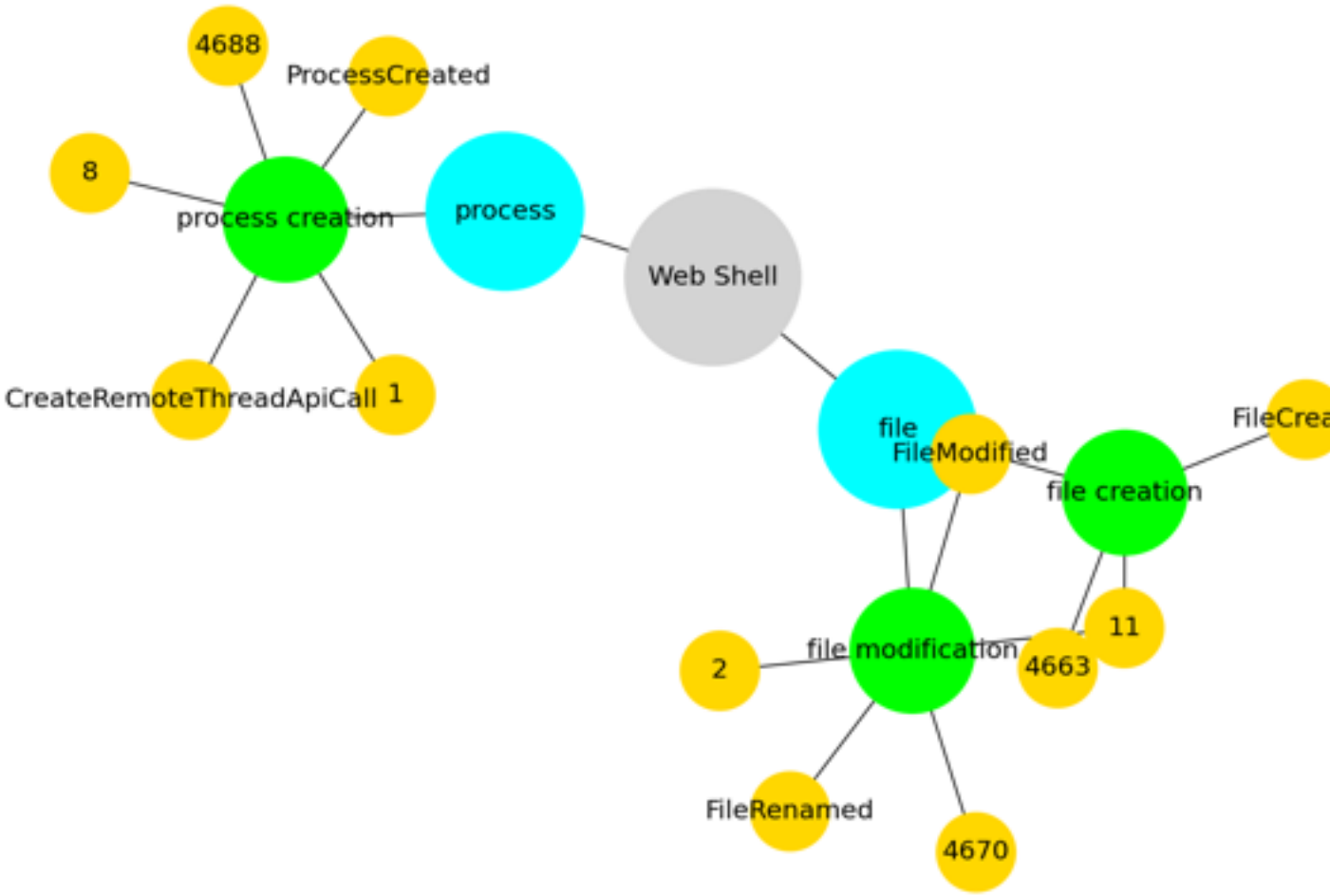
Used by group : BackdoorDiplomacy, GALLIUM, APT39, OilRig, APT29, Deep Panda

Tactic : persistence

Technique : Web Shell

Adversaries may backdoor web servers with web shells to establish persistent access to systems. A Web shell is a Web script that is placed on an openly accessible Web server to allow an adversary to use the Web server as a gateway into a network. A Web shell may provide a set of functions to execute or a command-line interface on the system that hosts the Web server.

In addition to a server-side script, a Web shell may have a client interface program that is used to talk to the Web server (ex: [China Chopper](https://attack.mitre.org/software/S0020) Web shell client).(Citation: Lee 2013)



3.12 T1036.005

Used by group : BackdoorDiplomacy, APT41, Machete, APT39, MuddyWater, APT29

Tactic : defense-evasion

Technique : Match Legitimate Name or Location

Adversaries may match or approximate the name or location of legitimate files or resources when naming/placing them. This is done for the sake of evading defenses and observation. This may be done by placing an executable in a commonly trusted directory (ex: under System32) or giving it the name of a legitimate, trusted program (ex: svchost.exe). In containerized environments, this may also be done by creating a resource in a namespace that matches the naming convention of a container pod or cluster. Alternatively, a file or container image name given may be a close approximation to legitimate programs/images or something innocuous.

Adversaries may also use the same icon of the file they are trying to mimic.

3.13 T1033

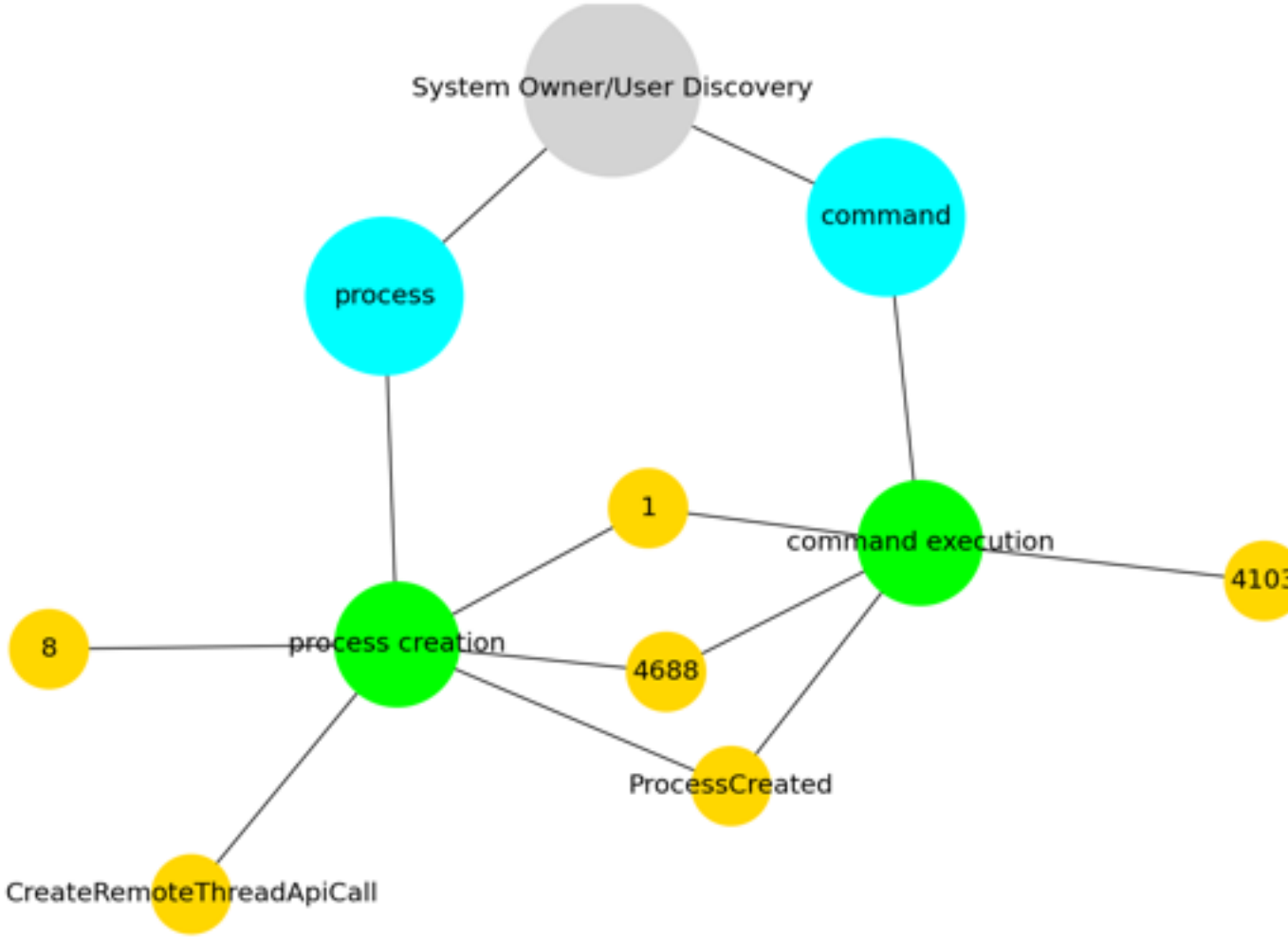
Used by group : APT41, GALLIUM, APT39, APT19, MuddyWater, OilRig

Tactic : discovery

Technique : System Owner/User Discovery

Adversaries may attempt to identify the primary user, currently logged in user, set of users that commonly uses a system, or whether a user is actively using the system. They may do this, for example, by retrieving account usernames or by using [OS Credential Dumping](https://attack.mitre.org/techniques/T1003). The information may be collected in a number of different ways using other Discovery techniques, because user and username details are prevalent throughout a system and include running process ownership, file/directory ownership, session information, and system logs. Adversaries may use the information from [System Owner/User Discovery](https://attack.mitre.org/techniques/T1033) during automated discovery to shape follow-on behaviors, including whether or not the adversary fully infects the target and/or attempts specific actions.

Various utilities and commands may acquire this information, including <code>whoami</code>. In macOS and Linux, the currently logged in user can be identified with <code>w</code> and <code>who</code>. On macOS the <code>dscl . list /Users | grep -v '_'</code> command can also be used to enumerate user accounts. Environment variables, such as <code>%USERNAME%</code> and <code>$USER</code>, may also be used to access this information.



3.14 T1003.001

Used by group : Aquatic Panda, APT41, GALLIUM, APT39, MuddyWater, OilRig

Tactic : credential-access

Technique : LSASS Memory

Adversaries may attempt to access credential material stored in the process memory of the Local Security Authority Subsystem Service (LSASS). After a user logs on, the system generates and stores a variety of credential materials in LSASS process memory. These credential materials can be harvested by an administrative user or SYSTEM and used to conduct [Lateral Movement](https://attack.mitre.org/tactics/TA0008) using [Use Alternate Authentication Material](https://attack.mitre.org/techniques/T1550).

As well as in-memory techniques, the LSASS process memory can be dumped from the target host and analyzed on a local system.

For example, on the target host use procdump:

* <code>procdump -ma lsass.exe lsass_dump</code>

Locally, mimikatz can be run using:

* <code>sekurlsa::Minidump lsassdump.dmp</code>
* <code>sekurlsa::logonPasswords</code>

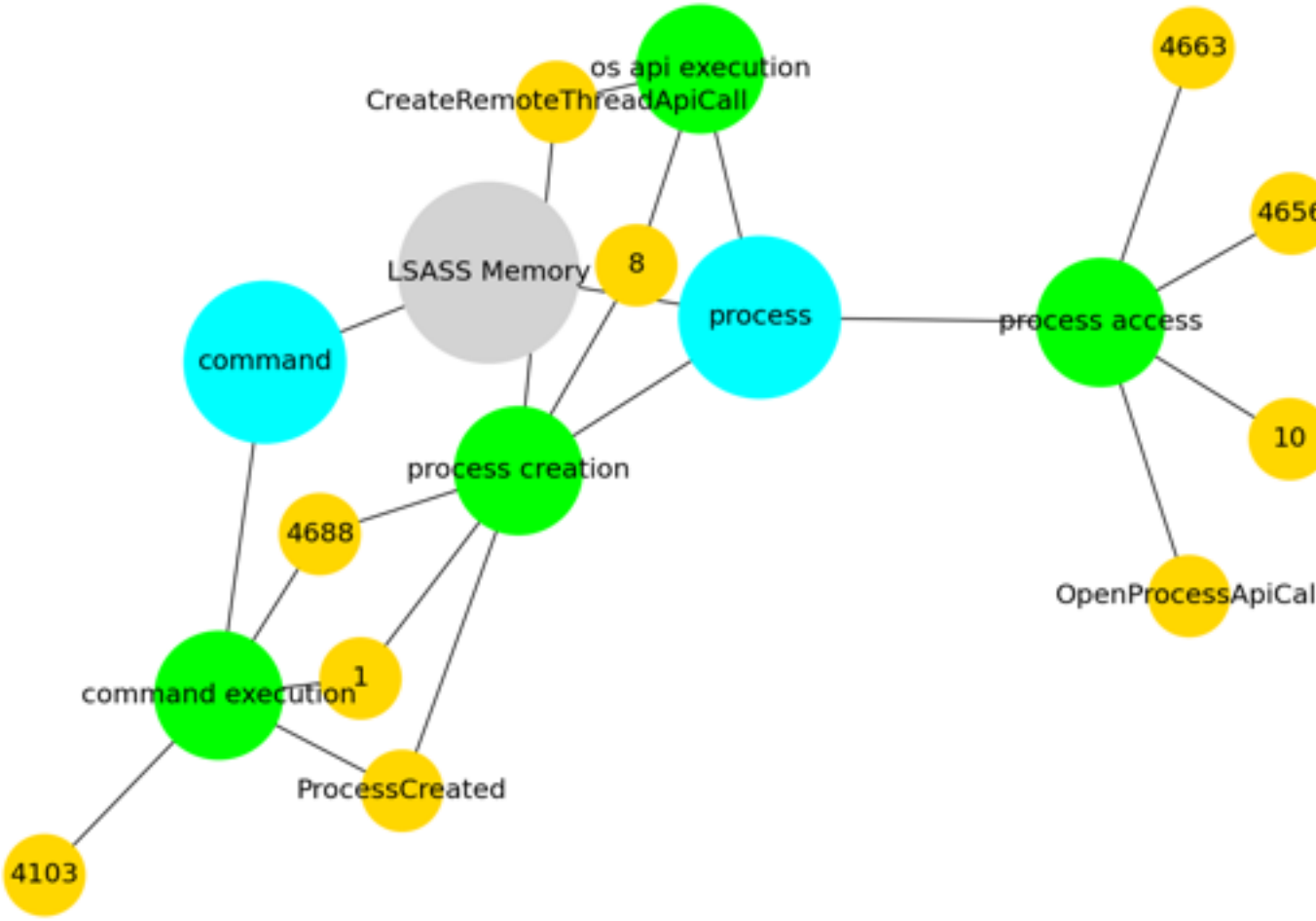
Built-in Windows tools such as comsvcs.dll can also be used:

* <code>rundll32.exe C:\Windows\System32\comsvcs.dll MiniDump PID lsass.dmp full</code>(Citation: Volexity Exchange Marauder March 2021)(Citation: Symantec Attacks Against Government Sector)

Windows Security Support Provider (SSP) DLLs are loaded into LSSAS process at system start. Once loaded into the LSA, SSP DLLs have access to encrypted and plaintext passwords that are stored in Windows, such as any logged-on user's Domain password or smart card PINs. The SSP configuration is stored in two Registry keys: <code>HKLM\SYSTEM\CurrentControlSet\Control\Lsa\Security Packages</code> and <code>HKLM\SYSTEM\CurrentControlSet\Control\Lsa\OSConfig\Security Packages</code>. An adversary may modify these Registry keys to add new SSPs, which will be loaded the next time the system boots, or when the AddSecurityPackage Windows API function is called.(Citation: Graeber 2014)

The following SSPs can be used to access credentials:

* Msv: Interactive logons, batch logons, and service logons are done through the MSV authentication package.
* Wdigest: The Digest Authentication protocol is designed for use with Hypertext Transfer Protocol (HTTP) and Simple Authentication Security Layer (SASL) exchanges.(Citation: TechNet Blogs Credential Protection)
* Kerberos: Preferred for mutual client-server domain authentication in Windows 2000 and later.
* CredSSP: Provides SSO and Network Level Authentication for Remote Desktop Services.(Citation: TechNet Blogs Credential Protection)



3.15 T1560.001

Used by group : Aquatic Panda, APT41, GALLIUM, APT39, MuddyWater, APT29

Tactic : collection

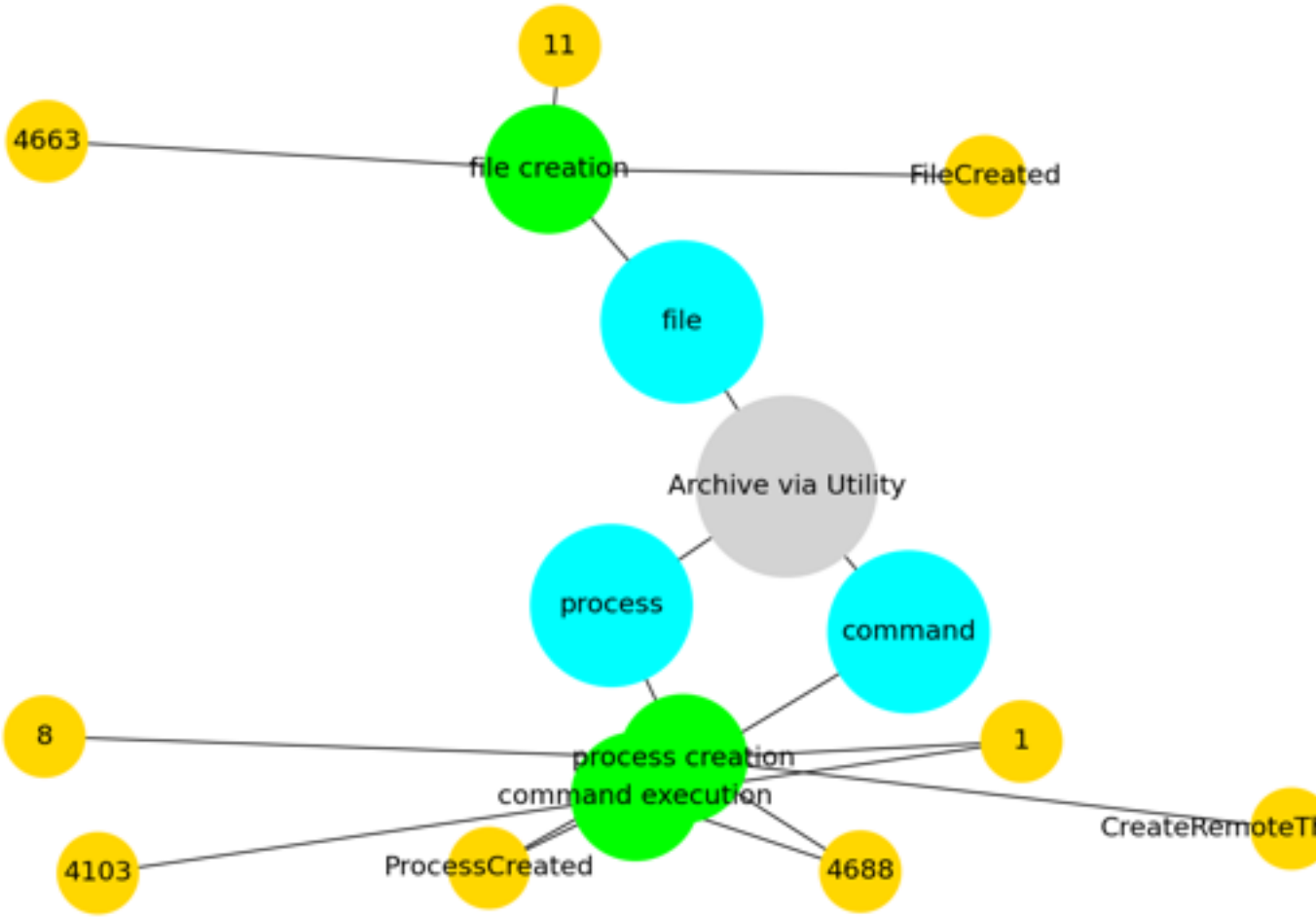
Technique : Archive via Utility

Adversaries may use utilities to compress and/or encrypt collected data prior to exfiltration. Many utilities include functionalities to compress, encrypt, or otherwise package data into a format that is easier/more secure to transport.

Adversaries may abuse various utilities to compress or encrypt data before exfiltration. Some third party utilities may be preinstalled, such as <code>tar</code> on Linux and macOS or <code>zip</code> on Windows systems. On Windows, <code>diantz</code> or <code> makecab</code> may be used to package collected files into a cabinet (.cab) file. <code>diantz</code> may also be used to download and compress files from remote locations (i.e. [Remote Data Staging](https://attack.mitre.org/techniques/T1074/002)).(Citation: diantz.exe_lolbas) Additionally, <code>xcopy</code> on Windows

can copy files and directories with a variety of options.

Adversaries may use also third party utilities, such as 7-Zip, WinRAR, and WinZip, to perform similar activities.(Citation: 7zip Homepage)(Citation: WinRAR Homepage)(Citation: WinZip Homepage)



3.16 T1082

Used by group : Aquatic Panda, APT19, MuddyWater, OilRig, APT29

Tactic : discovery

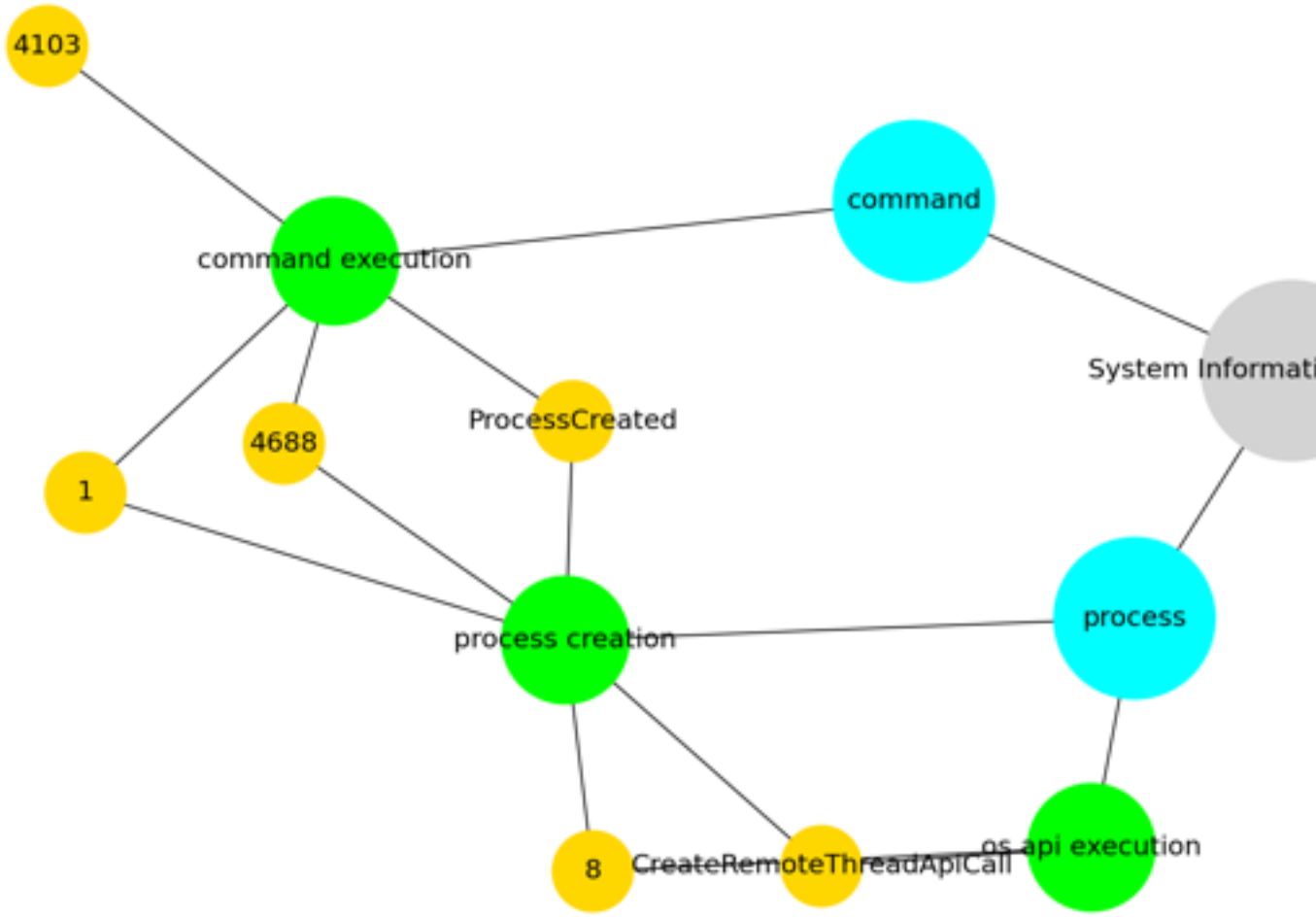
Technique : System Information Discovery

An adversary may attempt to get detailed information about the operating system and hardware, including version, patches, hotfixes, service packs, and architecture. Adversaries may use the information from [System Information Discovery](https://attack.mitre.org/techniques/T1082) during automated discovery to shape follow-on behaviors, including whether or

not the adversary fully infects the target and/or attempts specific actions.

Tools such as [Systeminfo](https://attack.mitre.org/software/S0096) can be used to gather detailed system information. If running with privileged access, a breakdown of system data can be gathered through the <code>systemsetup</code> configuration tool on macOS. As an example, adversaries with user-level access can execute the <code>df -aH</code> command to obtain currently mounted disks and associated freely available space. Adversaries may also leverage a [Network Device CLI](https://attack.mitre.org/techniques/T1059/008) on network devices to gather detailed system information.(Citation: US-CERT-TA18-106A) [System Information Discovery](https://attack.mitre.org/techniques/T1082) combined with information gathered from other forms of discovery and reconnaissance can drive payload development and concealment.(Citation: OSX.FairyTale)(Citation: 20 macOS Common Tools and Techniques)

Infrastructure as a Service (IaaS) cloud providers such as AWS, GCP, and Azure allow access to instance and virtual machine information via APIs. Successful authenticated API calls can return data such as the operating system platform and status of a particular instance or the model view of a virtual machine.(Citation: Amazon Describe Instance)(Citation: Google Instances Resource)(Citation: Microsoft Virutal Machine API)



3.17 T1070.004

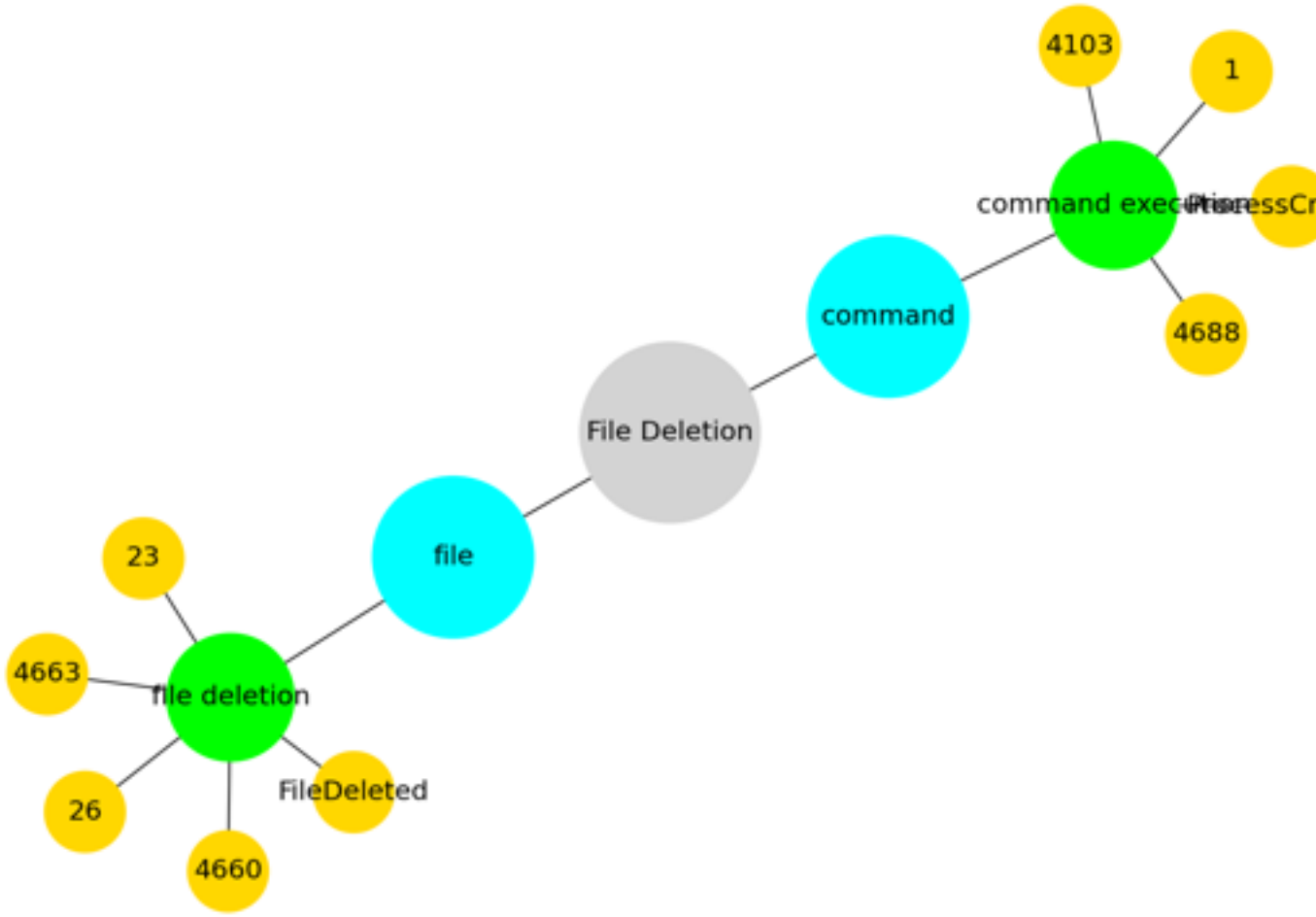
Used by group : Aquatic Panda, APT41, APT39, OilRig, APT29

Tactic : defense-evasion

Technique : File Deletion

Adversaries may delete files left behind by the actions of their intrusion activity. Malware, tools, or other non-native files dropped or created on a system by an adversary (ex: [Ingress Tool Transfer](https://attack.mitre.org/techniques/T1105)) may leave traces to indicate to what was done within a network and how. Removal of these files can occur during an intrusion, or as part of a post-intrusion process to minimize the adversary's footprint.

There are tools available from the host operating system to perform cleanup, but adversaries may use other tools as well.(Citation: Microsoft SDelete July 2016) Examples of built-in [Command and Scripting Interpreter](https://attack.mitre.org/techniques/T1059) functions include <code>del</code> on Windows and <code>rm</code> or <code>unlink</code> on Linux and macOS.



3.18 T1566.002

Used by group : Machete, APT39, MuddyWater, OilRig, APT29

Tactic : initial-access

Technique : Spearphishing Link

Adversaries may send spearphishing emails with a malicious link in an attempt to gain access to victim systems. Spearphishing with a link is a specific variant of spearphishing. It is different from other forms of spearphishing in that it employs the use of links to download malware contained in email, instead of attaching malicious files to the email itself, to avoid defenses that may inspect email attachments. Spearphishing may also involve social engineering techniques, such as posing as a trusted source.

All forms of spearphishing are electronically delivered social engineering targeted at a specific individual, company, or industry. In this case, the malicious emails contain links. Generally, the links will be accompanied by social engineering text and require the user to actively click or copy and paste a URL into a browser, leveraging [User Execution](https://attack.mitre.org/techniques/T1204). The

visited website may compromise the web browser using an exploit, or the user will be prompted to download applications, documents, zip files, or even executables depending on the pretext for the email in the first place. Adversaries may also include links that are intended to interact directly with an email reader, including embedded images intended to exploit the end system directly or verify the receipt of an email (i.e. web bugs/web beacons).

Adversaries may also utilize links to perform consent phishing, typically with OAuth 2.0 request URLs that when accepted by the user provide permissions/access for malicious applications, allowing adversaries to [Steal Application Access Token](https://attack.mitre.org/techniques/T1528)s.(Citation: Trend Micro Pawn Storm OAuth 2017) These stolen access tokens allow the adversary to perform various actions on behalf of the user via API calls. (Citation: Microsoft OAuth 2.0 Consent Phishing 2021)

3.19 T1140

Used by group : APT39, APT19, MuddyWater, OilRig, APT29

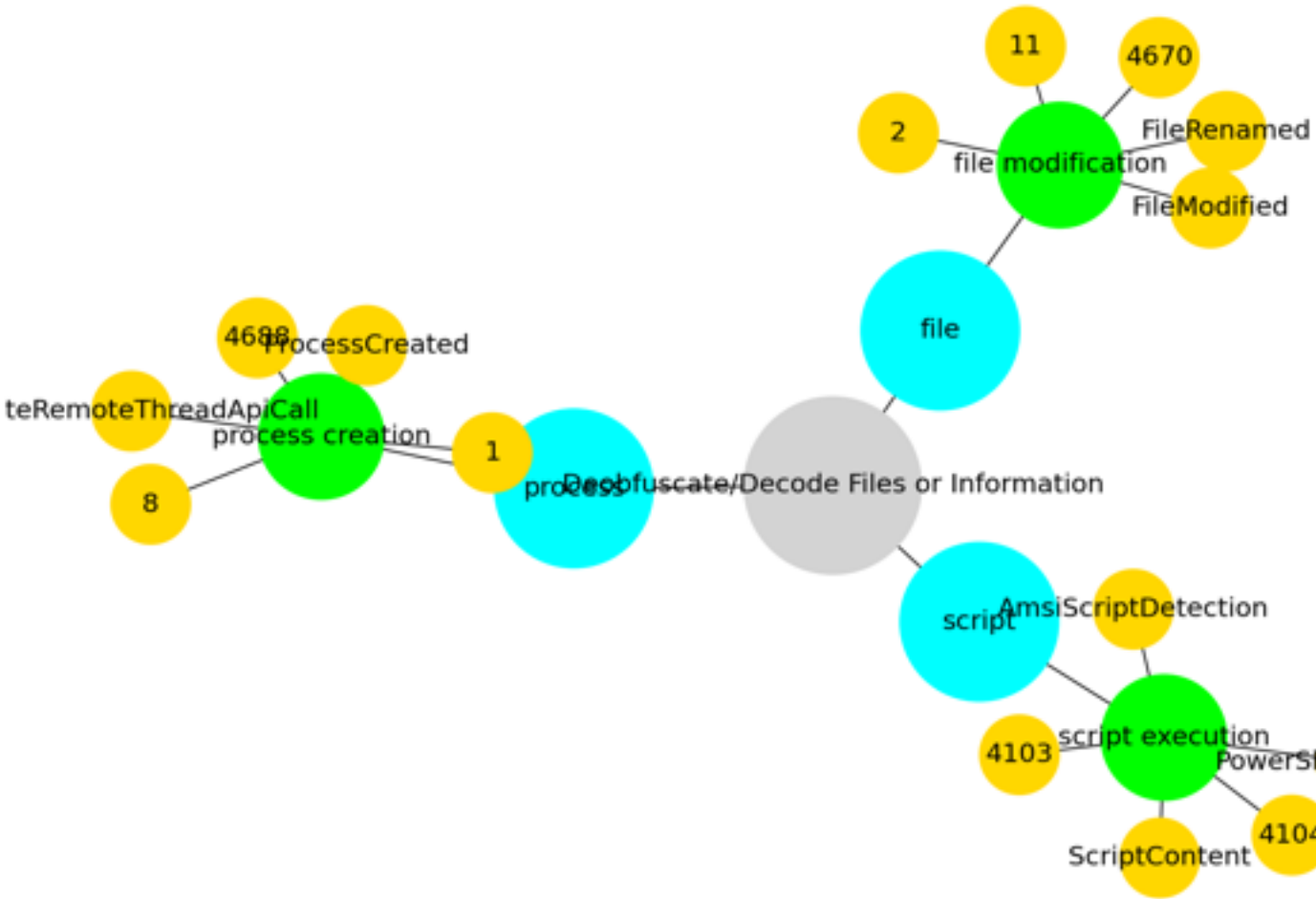
Tactic : defense-evasion

Technique : Deobfuscate/Decode Files or Information

Adversaries may use [Obfuscated Files or Information](https://attack.mitre.org/techniques/T1027) to hide artifacts of an intrusion from analysis. They may require separate mechanisms to decode or deobfuscate that information depending on how they intend to use it. Methods for doing that include built-in functionality of malware or by using utilities present on the system.

One such example is use of [certutil](https://attack.mitre.org/software/S0160) to decode a remote access tool portable executable file that has been hidden inside a certificate file. (Citation: Malwarebytes Targeted Attack against Saudi Arabia) Another example is using the Windows <code>copy /b</code> command to reassemble binary fragments into a malicious payload. (Citation: Carbon Black Obfuscation Sept 2016)

Sometimes a user's action may be required to open it for deobfuscation or decryption as part of [User Execution](https://attack.mitre.org/techniques/T1204). The user may also be required to input a password to open a password protected compressed/encrypted file that was provided by the adversary. (Citation: Volexity PowerDuke November 2016)



3.20 T1078

Used by group : APT41, GALLIUM, APT39, OilRig, APT29

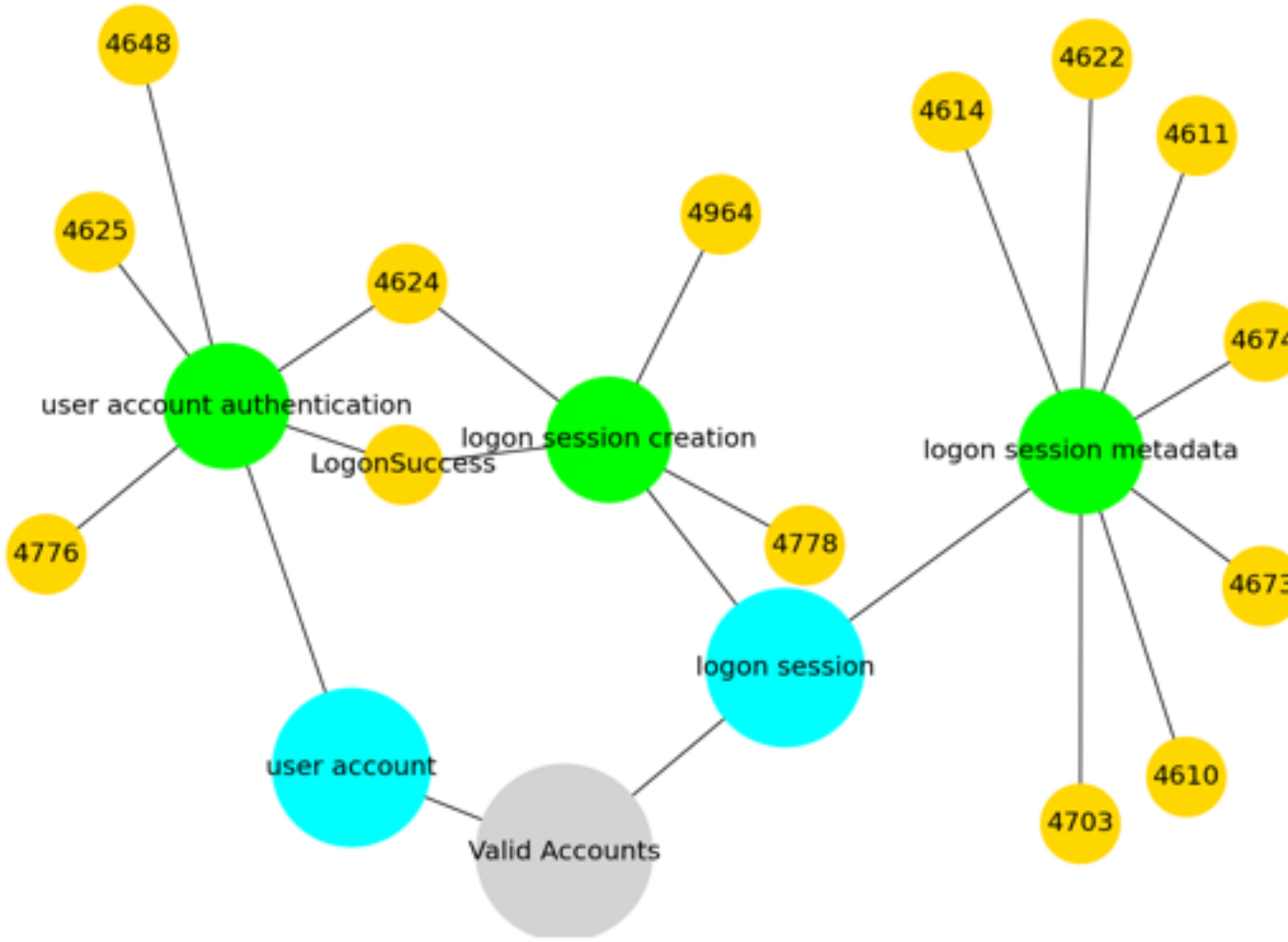
Tactic : defense-evasion, persistence, privilege-escalation, initial-access

Technique : Valid Accounts

Adversaries may obtain and abuse credentials of existing accounts as a means of gaining Initial Access, Persistence, Privilege Escalation, or Defense Evasion. Compromised credentials may be used to bypass access controls placed on various resources on systems within the network and may even be used for persistent access to remote systems and externally available services, such as VPNs, Outlook Web Access and remote desktop. Compromised credentials may also grant an adversary increased privilege to specific systems or access to restricted areas of the network. Adversaries may choose not to use malware or tools in conjunction with the legitimate access those credentials provide to make it harder to detect their presence.

In some cases, adversaries may abuse inactive accounts: for example, those belonging to individuals who are no longer part of an organization. Using these accounts may allow the adversary to evade detection, as the original account user will not be present to identify any anomalous activity taking place on their account.(Citation: CISA MFA PrintNightmare)

The overlap of permissions for local, domain, and cloud accounts across a network of systems is of concern because the adversary may be able to pivot across accounts and systems to reach a high level of access (i.e., domain or enterprise administrator) to bypass access controls set within the enterprise.(Citation: TechNet Credential Theft)



3.21 T1049

Used by group : BackdoorDiplomacy, APT41, GALLIUM, MuddyWater, OilRig

Tactic : discovery

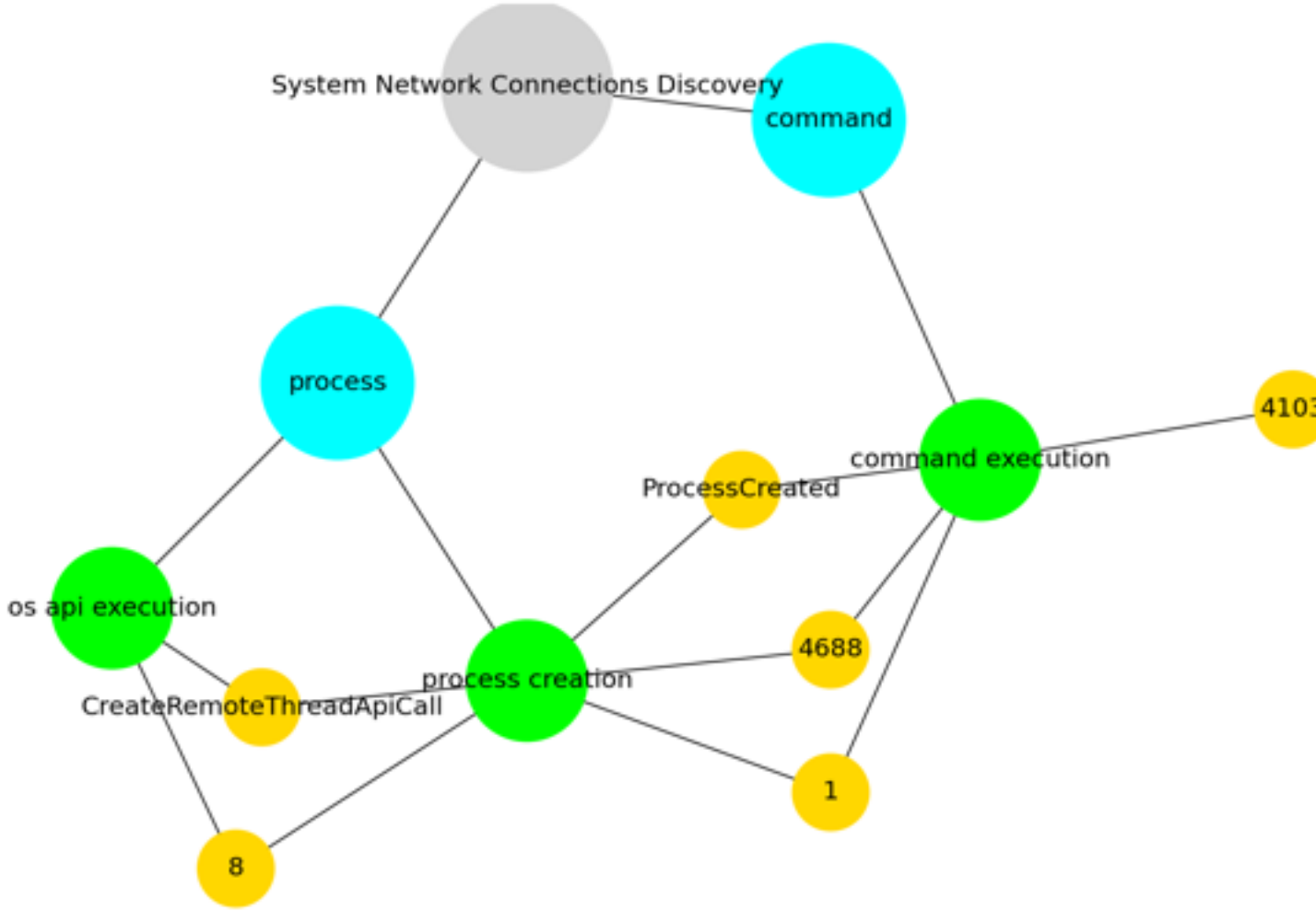
Technique : System Network Connections Discovery

Adversaries may attempt to get a listing of network connections to or from the compromised system they are currently accessing or from remote systems by querying for information over the network.

An adversary who gains access to a system that is part of a cloud-based environment may map out Virtual Private Clouds or Virtual Networks in order to determine what systems and services are connected. The actions performed are likely the same types of discovery techniques depending on the operating system, but the resulting information may include details about the networked cloud environment relevant to the adversary's goals. Cloud providers may have different ways in which their virtual networks operate.(Citation: Amazon AWS VPC Guide)(Citation: Microsoft Azure Virtual Network Overview)(Citation: Google VPC Overview)

Similarly, adversaries who gain access to network devices may also perform similar discovery activities to gather information about connected systems and services.

Utilities and commands that acquire this information include [netstat](https://attack.mitre.org/software/S0104), "net use," and "net session" with [Net](https://attack.mitre.org/software/S0039). In Mac and Linux, [netstat](https://attack.mitre.org/software/S0104) and <code>lsof</code> can be used to list current connections. <code>who -a</code> and <code>w</code> can be used to show which users are currently logged in, similar to "net session". Additionally, built-in features native to network devices and [Network Device CLI](https://attack.mitre.org/techniques/T1059/008) may be used.(Citation: US-CERT-TA18-106A)



3.22 T1059.005

Used by group : Machete, APT39, MuddyWater, OilRig, APT29

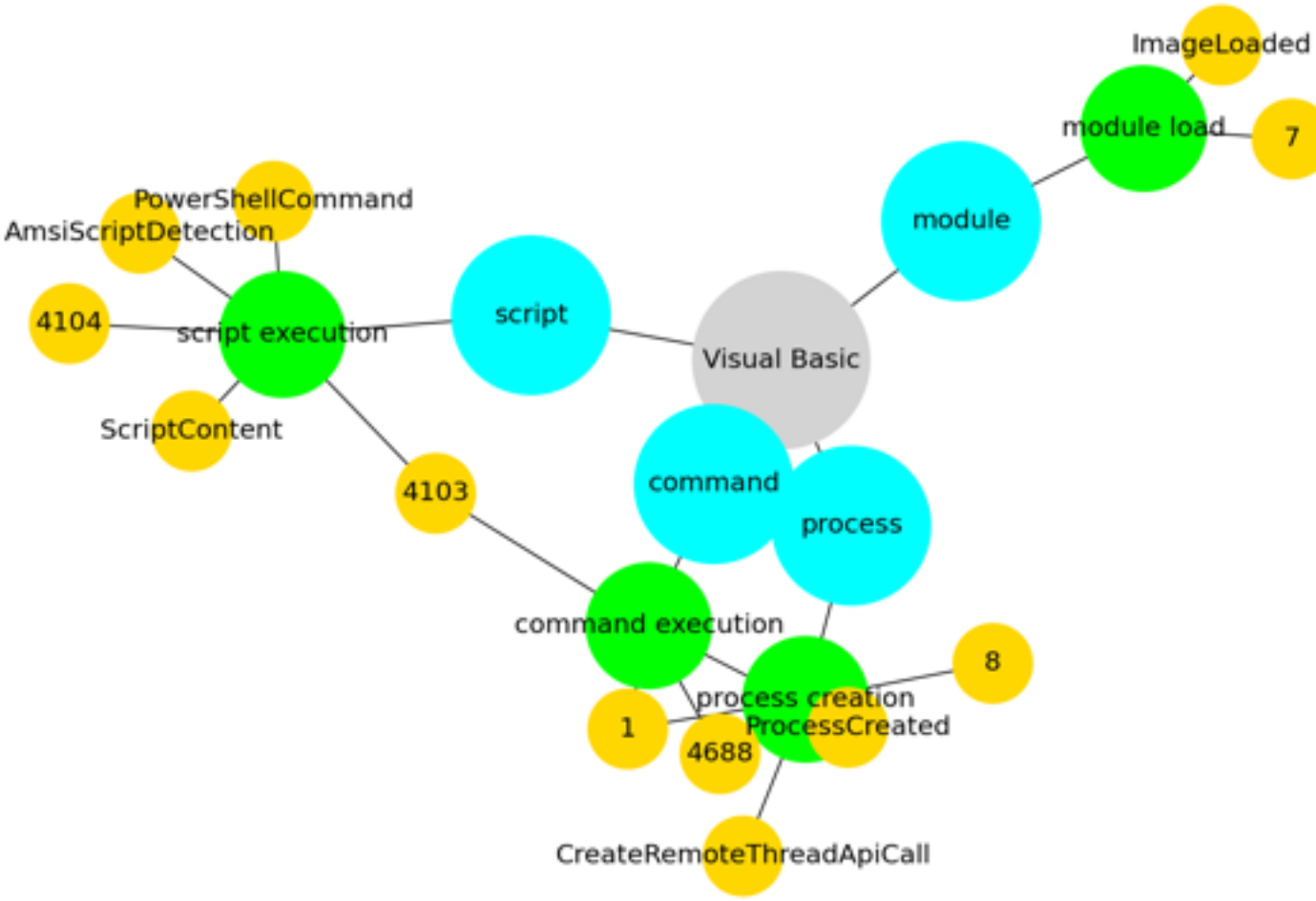
Tactic : execution

Technique : Visual Basic

Adversaries may abuse Visual Basic (VB) for execution. VB is a programming language created by Microsoft with interoperability with many Windows technologies such as [Component Object Model](https://attack.mitre.org/techniques/T1559/001) and the [Native API](https://attack.mitre.org/techniques/T1106) through the Windows API. Although tagged as legacy with no planned future evolutions, VB is integrated and supported in the .NET Framework and cross-platform .NET Core.(Citation: VB .NET Mar 2020)(Citation: VB Microsoft)

Derivative languages based on VB have also been created, such as Visual Basic for Applications (VBA) and VBScript. VBA is an event-driven programming language built into Microsoft Office, as well as several third-party applications.(Citation: Microsoft VBA)(Citation: Wikipedia VBA) VBA enables documents to contain macros used to automate the execution of tasks and other functionality on the host. VBScript is a default scripting language on Windows hosts and can also be used in place of [JavaScript](https://attack.mitre.org/techniques/T1059/007) on HTML Application (HTA) webpages served to Internet Explorer (though most modern browsers do not come with VBScript support).(Citation: Microsoft VBScript)

Adversaries may use VB payloads to execute malicious commands. Common malicious usage includes automating execution of behaviors with VBScript or embedding VBA content into [Spearphishing Attachment](https://attack.mitre.org/techniques/T1566/001) payloads (which may also involve [Mark-of-the-Web Bypass](https://attack.mitre.org/techniques/T1553/005) to enable execution).(Citation: Default VBS macros Blocking)



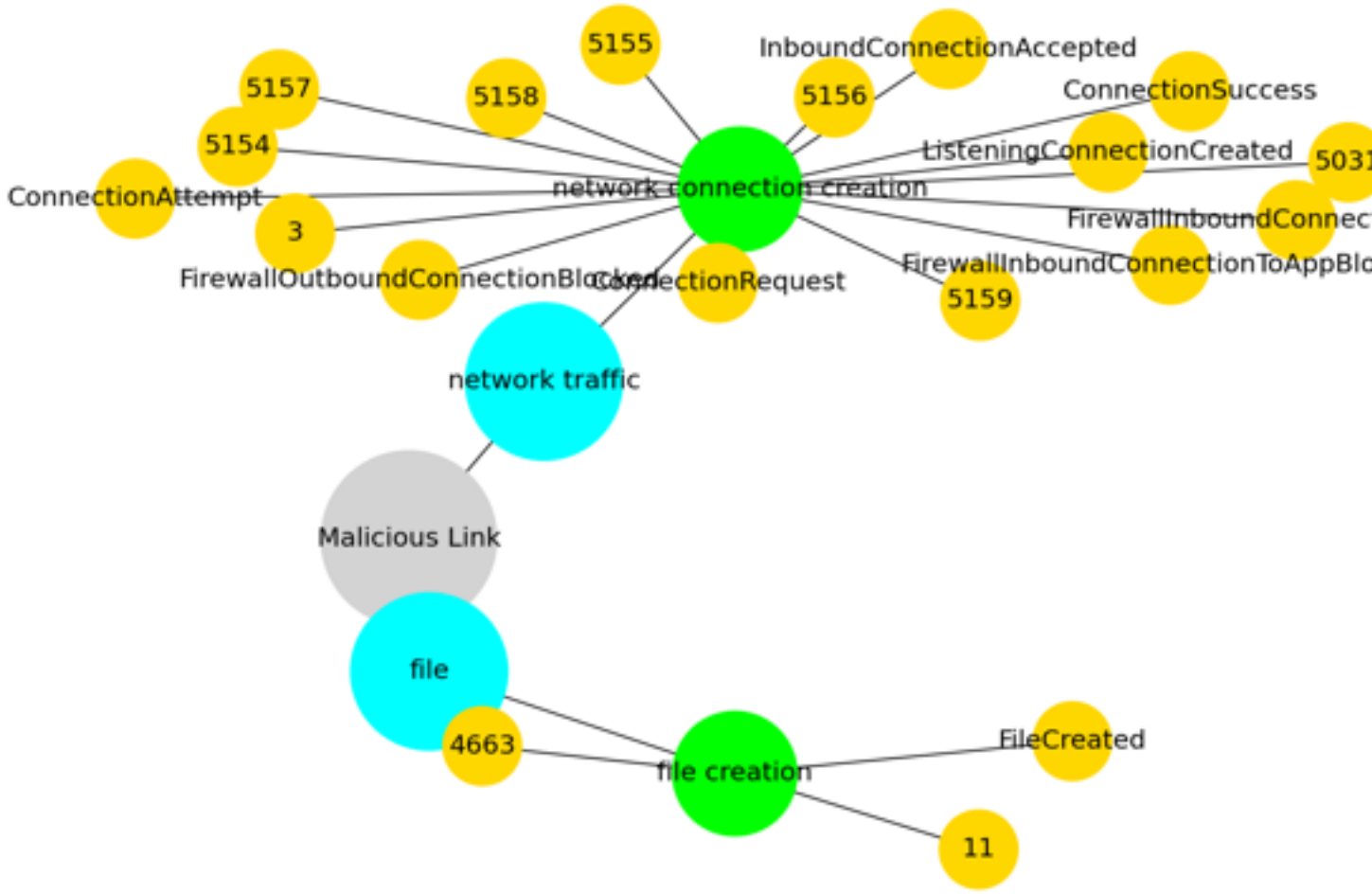
3.23 T1204.001

Used by group : Machete, APT39, MuddyWater, OilRig, APT29

Tactic : execution

Technique : Malicious Link

An adversary may rely upon a user clicking a malicious link in order to gain execution. Users may be subjected to social engineering to get them to click on a link that will lead to code execution. This user action will typically be observed as follow-on behavior from [Spearphishing Link](https://attack.mitre.org/techniques/T1566/002). Clicking on a link may also lead to other execution techniques such as exploitation of a browser or application vulnerability via [Exploitation for Client Execution](https://attack.mitre.org/techniques/T1203). Links may also lead users to download files that require execution via [Malicious File](https://attack.mitre.org/techniques/T1204/002).



3.24 T1190

Used by group : BackdoorDiplomacy, APT41, GALLIUM, APT39, APT29

Tactic : initial-access

Technique : Exploit Public-Facing Application

Adversaries may attempt to take advantage of a weakness in an Internet-facing computer or program using software, data, or commands in order to cause unintended or unanticipated behavior. The weakness in the system can be a bug, a glitch, or a design vulnerability. These applications are often websites, but can include databases (like SQL), standard services (like SMB or SSH), network device administration and management protocols (like SNMP and Smart Install), and any other applications with Internet accessible open sockets, such as web servers and related services.(Citation: NVD CVE-2016-6662)(Citation: CIS Multiple SMB Vulnerabilities)(Citation: US-CERT TA18-106A Network Infrastructure Devices 2018)(Citation: Cisco Blog Legacy Device Attacks)(Citation: NVD CVE-2014-7169) Depending on the flaw being exploited this may include [Exploitation for Defense Evasion](https://attack.mitre.org/techniques/T1211).

If an application is hosted on cloud-based infrastructure and/or is containerized, then exploiting it may lead to compromise of the underlying instance or container. This can allow an adversary a path to access the cloud or container APIs, exploit container host access via [Escape to Host](https://attack.mitre.org/techniques/T1611), or take advantage of weak identity and access management policies.

For websites and databases, the OWASP top 10 and CWE top 25 highlight the most common web-based vulnerabilities.(Citation: OWASP Top 10)(Citation: CWE top 25)

3.25 T1547.001

Used by group : APT41, APT39, APT19, MuddyWater, APT29

Tactic : persistence, privilege-escalation

Technique : Registry Run Keys / Startup Folder

Adversaries may achieve persistence by adding a program to a startup folder or referencing it with a Registry run key. Adding an entry to the "run keys" in the Registry or startup folder will cause the program referenced to be executed when a user logs in.(Citation: Microsoft Run Key) These programs will be executed under the context of the user and will have the account's associated permissions level.

Placing a program within a startup folder will also cause that program to execute when a user logs in. There is a startup folder location for individual user accounts as well as a system-wide startup folder that will be checked regardless of which user account logs in. The startup folder path for the current user is <code>C:\Users\\[Username]\AppData\Roaming\Microsoft\Windows\Start Menu\Programs\Startup</code>. The startup folder path for all users is <code>C:\ProgramData\Microsoft\Windows\Start Menu\Programs\StartUp</code>.

The following run keys are created by default on Windows systems:

* <code>HKEY_CURRENT_USER\Software\Microsoft\Windows\CurrentVersion\Run</code>
* <code>HKEY_CURRENT_USER\Software\Microsoft\Windows\CurrentVersion\RunOnce</code>
* <code>HKEY_LOCAL_MACHINE\Software\Microsoft\Windows\CurrentVersion\Run</code>
* <code>HKEY_LOCAL_MACHINE\Software\Microsoft\Windows\CurrentVersion\RunOnce</code>

Run keys may exist under multiple hives.(Citation: Microsoft Wow6432Node 2018)(Citation: Malwarebytes Wow6432Node 2016) The <code>HKEY_LOCAL_MACHINE\Software\Microsoft\Windows\CurrentVersion\RunOnceEx</code> is also available but is not created by default on Windows Vista and newer. Registry run key entries can reference programs directly or list them as a dependency.(Citation: Microsoft Run Key) For example, it is possible to load a DLL at logon using a "Depend" key with RunOnceEx: <code>reg add HKLM\SOFTWARE\Microsoft\Windows\CurrentVersion\RunOnceEx\0001\Depend /v 1 /d "C:\temp\evil[.]dll"</code> (Citation: Oddvar Moe RunOnceEx Mar 2018)

The following Registry keys can be used to set startup folder items for persistence:

* <code>HKEY_CURRENT_USER\Software\Microsoft\Windows\CurrentVersion\Explorer\User Shell Folders</code>
* <code>HKEY_CURRENT_USER\Software\Microsoft\Windows\CurrentVersion\Explorer\Shell Folders</code>
* <code>HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows\CurrentVersion\Explorer\Shell Folders</code>
* <code>HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows\CurrentVersion\Explorer\User Shell Folders</code>

The following Registry keys can control automatic startup of services during boot:

* <code>HKEY_LOCAL_MACHINE\Software\Microsoft\Windows\CurrentVersion\RunServicesOnce</code>
* <code>HKEY_CURRENT_USER\Software\Microsoft\Windows\CurrentVersion\RunServicesOnce</code>
* <code>HKEY_LOCAL_MACHINE\Software\Microsoft\Windows\CurrentVersion\RunServices</code>
* <code>HKEY_CURRENT_USER\Software\Microsoft\Windows\CurrentVersion\RunServices</code>

Using policy settings to specify startup programs creates corresponding values in either of two Registry keys:

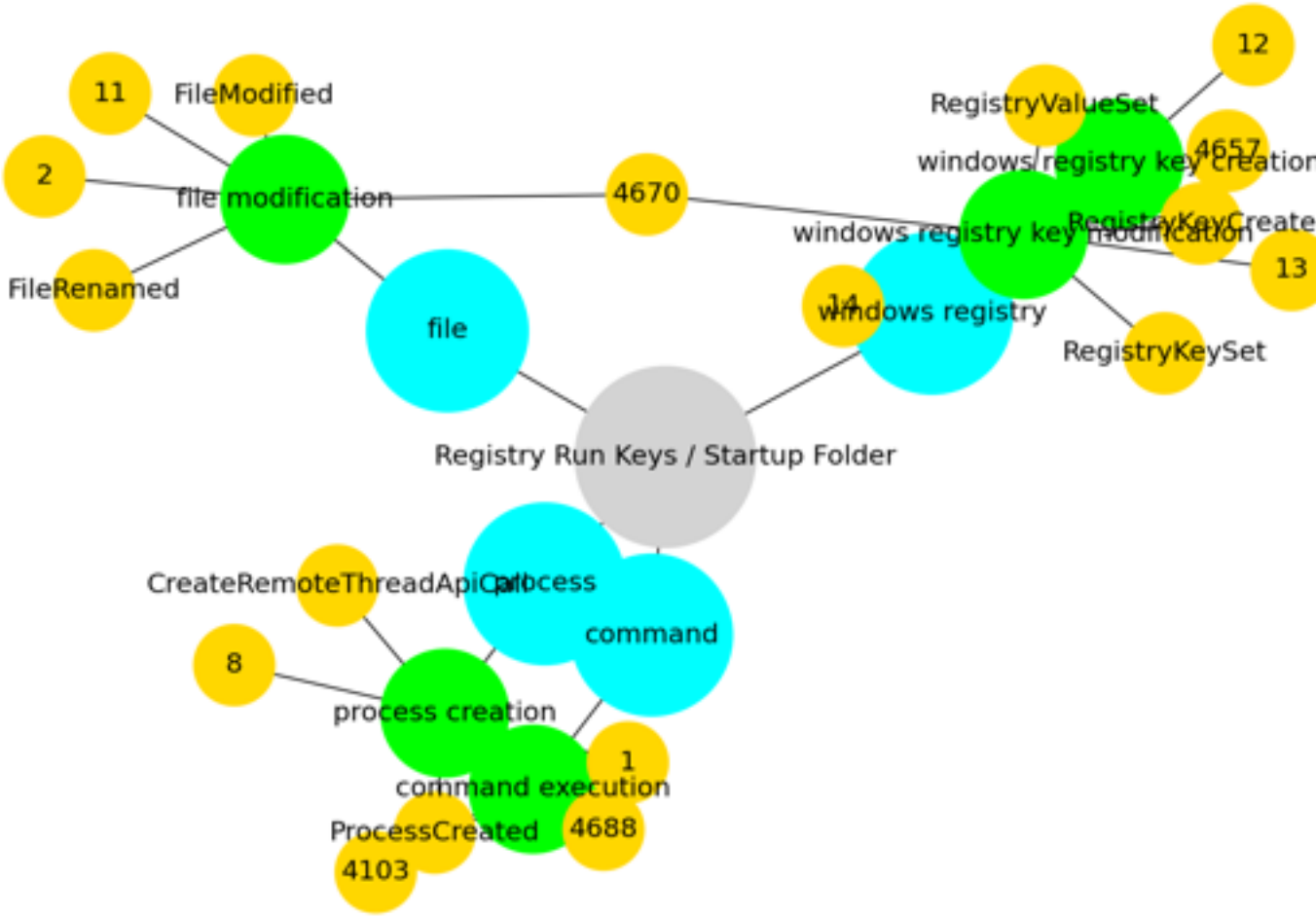
* <code>HKEY_LOCAL_MACHINE\Software\Microsoft\Windows\CurrentVersion\Policies\Explorer\Run</code>
* <code>HKEY_CURRENT_USER\Software\Microsoft\Windows\CurrentVersion\Policies\Explorer\Run</code>

The Winlogon key controls actions that occur when a user logs on to a computer running Windows 7. Most of these actions are under the control of the operating system, but you can also add custom actions here. The <code>HKEY_LOCAL_MACHINE\Software\Microsoft\Windows NT\CurrentVersion\Winlogon\Userinit</code> and <code>HKEY_LOCAL_MACHINE\Software\Microsoft\Windows NT\CurrentVersion\Winlogon\Shell</code> subkeys can automatically launch programs.

Programs listed in the load value of the registry key <code>HKEY_CURRENT_USER\Software\Microsoft\Windows NT\CurrentVersion\Windows</code> run when any user logs on.

By default, the multistring <code>BootExecute</code> value of the registry key <code>HKEY_LOCAL_MACHINE\System\CurrentControlSet\Control\Session Manager</code> is set to <code>autocheck autochk *</code>. This value causes Windows, at startup, to check the file-system integrity of the hard disks if the system has been shut down abnormally. Adversaries can add other programs or processes to this registry value which will automatically launch at boot.

Adversaries can use these configuration locations to execute malware, such as remote access tools, to maintain persistence through system reboots. Adversaries may also use [Masquerading](https://attack.mitre.org/techniques/T1036) to make the Registry entries look as if they are associated with legitimate programs.



3.26 T1016

Used by group : APT41, GALLIUM, APT19, MuddyWater, OilRig

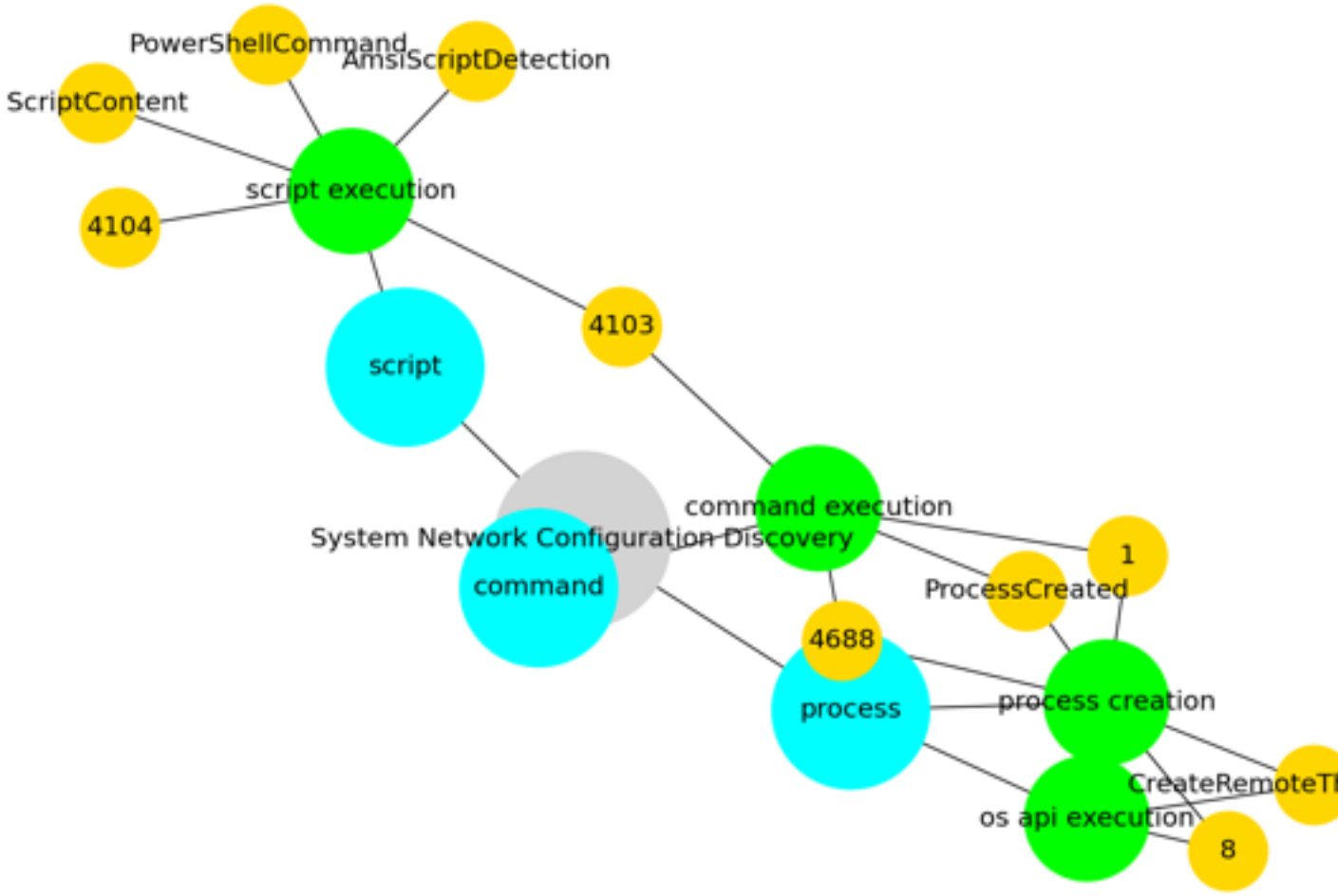
Tactic : discovery

Technique : System Network Configuration Discovery

Adversaries may look for details about the network configuration and settings, such as IP and/or MAC addresses, of systems they access or through information discovery of remote systems. Several operating system administration utilities exist that can be used to gather this information. Examples include [Arp](https://attack.mitre.org/software/S0099), [ipconfig](https://attack.mitre.org/software/S0100)/[ifconfig](https://attack.mitre.org/software/S0101), [nbtstat](https://attack.mitre.org/software/S0102), and [route](https://attack.mitre.org/software/S0103).

Adversaries may also leverage a [Network Device CLI](https://attack.mitre.org/techniques/T1059/008) on network devices to gather information about configurations and settings, such as IP addresses of configured interfaces and static/dynamic routes.(Citation: US-CERT-TA18-106A)(Citation: Mandiant APT41 Global Intrusion)

Adversaries may use the information from [System Network Configuration Discovery](https://attack.mitre.org/techniques/T1016) during automated discovery to shape follow-on behaviors, including determining certain access within the target network and what actions to do next.



3.27 T1083

Used by group : APT41, APT39, MuddyWater, APT29

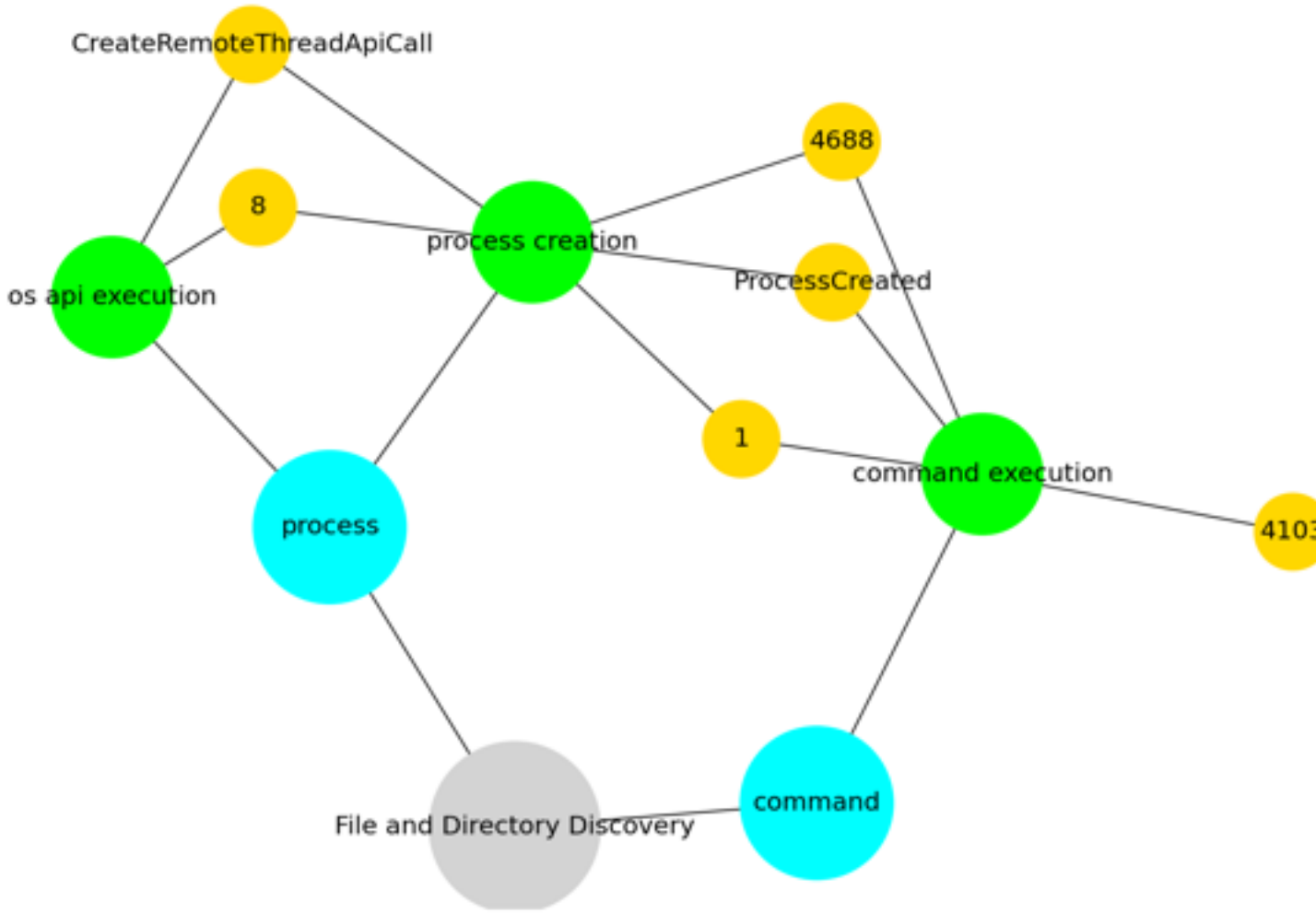
Tactic : discovery

Technique : File and Directory Discovery

Adversaries may enumerate files and directories or may search in specific locations of a host or network share for certain information within a file system. Adversaries may use the information from [File and Directory Discovery](https://attack.mitre.org/techniques/T1083) during automated discovery to shape follow-on behaviors, including whether or not the adversary fully infects the target and/or attempts specific actions.

Many command shell utilities can be used to obtain this information. Examples include <code>dir</code>, <code>tree</code>, <code>ls</code>, <code>find</code>, and <code>locate</code>.(Citation: Windows Commands JPCERT) Custom tools may also be used to gather file and directory information and interact with the [Native API](https://attack.mitre.org/techniques/T1106). Adversaries

may also leverage a [Network Device CLI](https://attack.mitre.org/techniques/T1059/008) on network devices to gather file and directory information.(Citation: US-CERT-TA18-106A)



3.28 T1018

Used by group : GALLIUM, APT39, APT29, Deep Panda

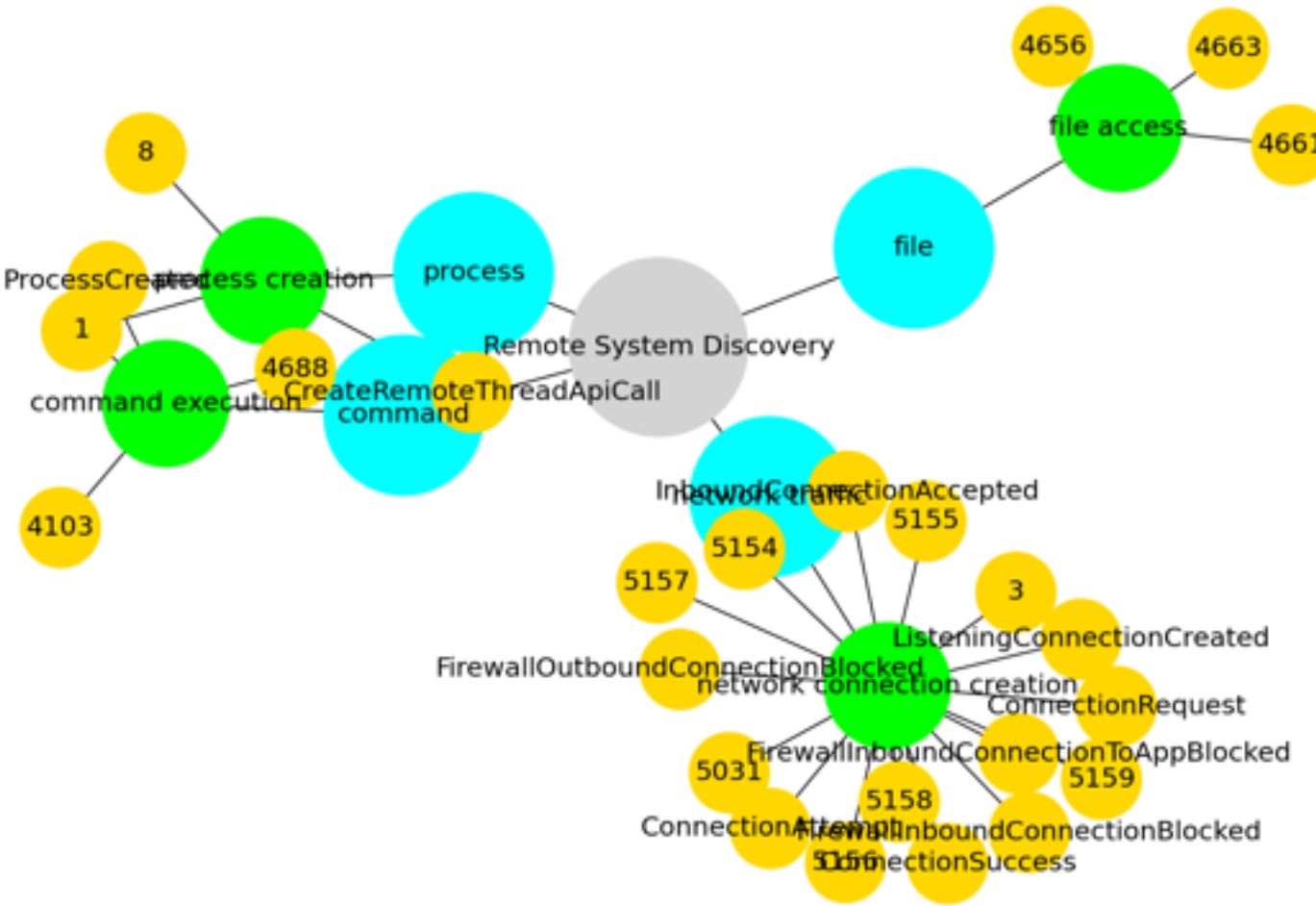
Tactic : discovery

Technique : Remote System Discovery

Adversaries may attempt to get a listing of other systems by IP address, hostname, or other logical identifier on a network that may be used for Lateral Movement from the current system. Functionality could exist within remote access tools to enable this, but utilities available on the operating system could also be used such as [Ping](https://attack.mitre.org/software/S0097) or <code>net view</code> using [Net](https://attack.mitre.org/software/S0039).

Adversaries may also analyze data from local host files (ex: <code>C:\Windows\System32\Drivers\etc\hosts</code> or <code>/etc/hosts</code>) or other passive means (such as local [Arp](https://attack.mitre.org/software/S0099) cache entries) in order to discover the presence of remote systems in an environment.

Adversaries may also target discovery of network infrastructure as well as leverage [Network Device CLI](https://attack.mitre.org/techniques/T1059/008) commands on network devices to gather detailed information about systems within a network.(Citation: US-CERT-TA18-106A)(Citation: CISA AR21-126A FIVEHANDS May 2021)



3.29 T1021.001

Used by group : APT41, APT39, OilRig, APT29

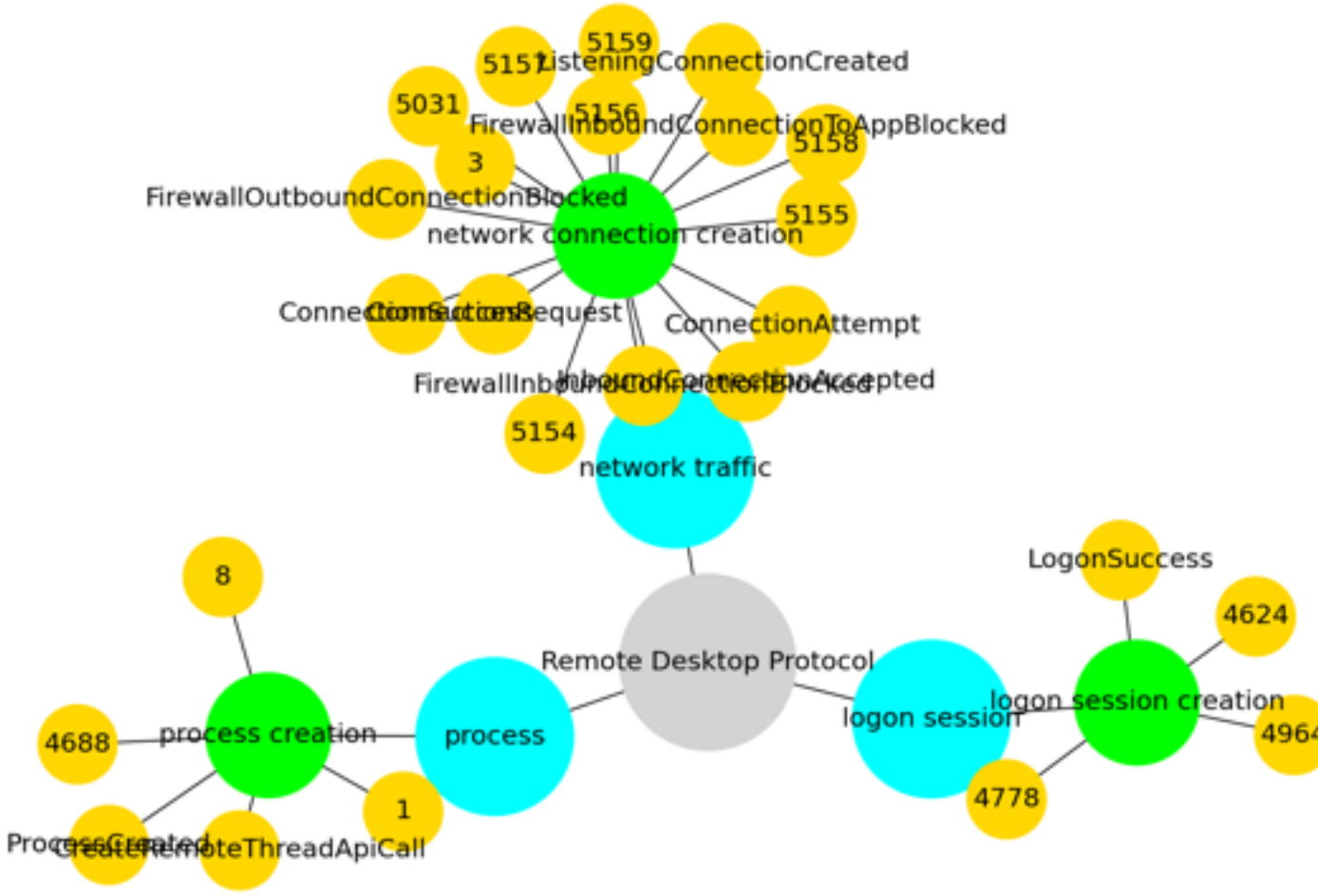
Tactic : lateral-movement

Technique : Remote Desktop Protocol

Adversaries may use [Valid Accounts](https://attack.mitre.org/techniques/T1078) to log into a computer using the Remote Desktop Protocol (RDP). The adversary may then perform actions as the logged-on user.

Remote desktop is a common feature in operating systems. It allows a user to log into an interactive session with a system desktop graphical user interface on a remote system. Microsoft refers to its implementation of the Remote Desktop Protocol (RDP) as Remote Desktop Services (RDS).(Citation: TechNet Remote Desktop Services)

Adversaries may connect to a remote system over RDP/RDS to expand access if the service is enabled and allows access to accounts with known credentials. Adversaries will likely use Credential Access techniques to acquire credentials to use with RDP. Adversaries may also use RDP in conjunction with the [Accessibility Features](https://attack.mitre.org/techniques/T1546/008) or [Terminal Services DLL](https://attack.mitre.org/techniques/T1505/005) for Persistence.(Citation: Alperovitch Malware)



3.30 T1021.002

Used by group : APT41, APT39, APT29, Deep Panda

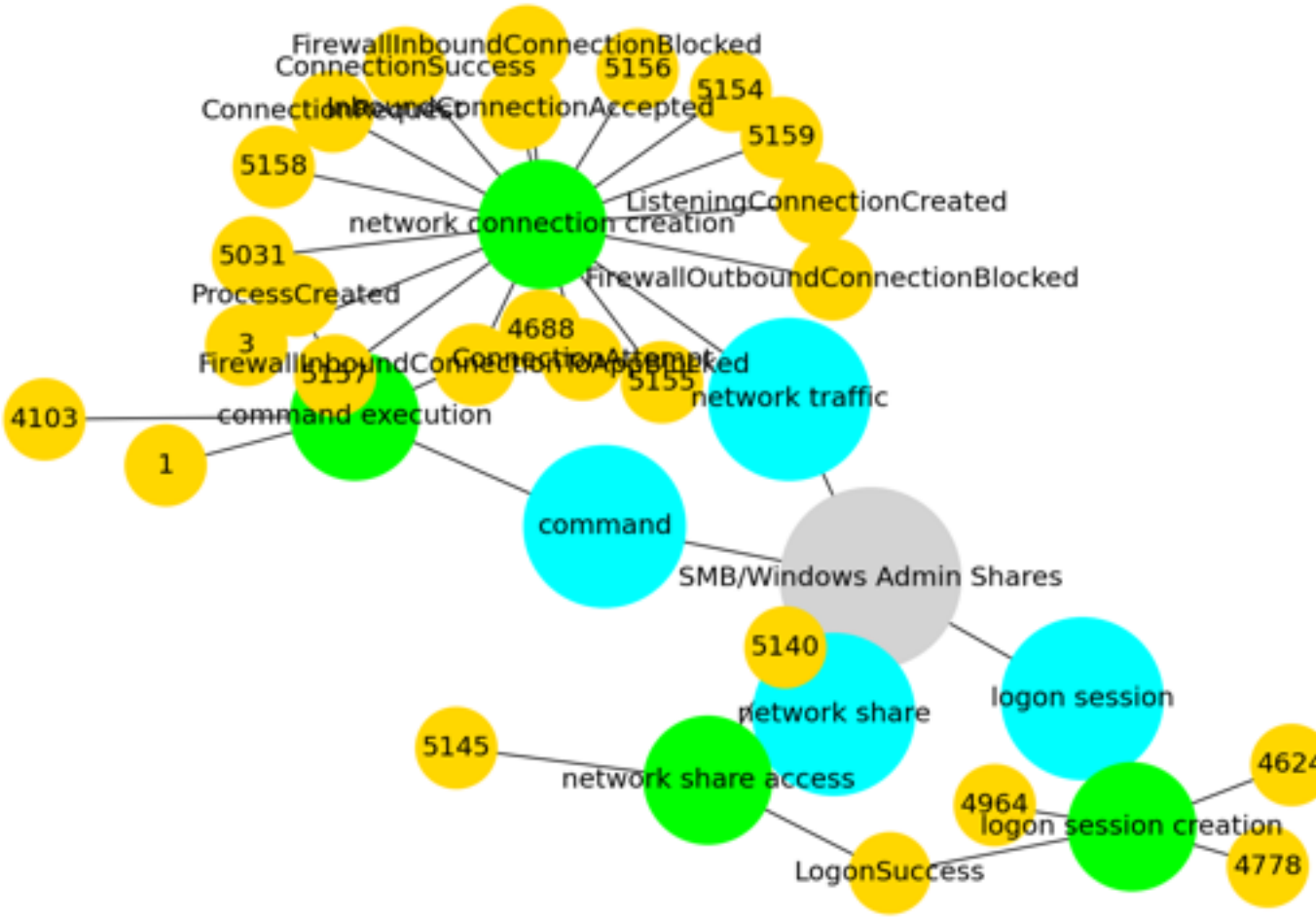
Tactic : lateral-movement

Technique : SMB/Windows Admin Shares

Adversaries may use [Valid Accounts](https://attack.mitre.org/techniques/T1078) to interact with a remote network share using Server Message Block (SMB). The adversary may then perform actions as the logged-on user.

SMB is a file, printer, and serial port sharing protocol for Windows machines on the same network or domain. Adversaries may use SMB to interact with file shares, allowing them to move laterally throughout a network. Linux and macOS implementations of SMB typically use Samba.

Windows systems have hidden network shares that are accessible only to administrators and provide the ability for remote file copy and other administrative functions. Example network shares include `C$`, `ADMIN$`, and `IPC$`. Adversaries may use this technique in conjunction with administrator-level [Valid Accounts](https://attack.mitre.org/techniques/T1078) to remotely access a networked system over SMB,(Citation: Wikipedia Server Message Block) to interact with systems using remote procedure calls (RPCs),(Citation: TechNet RPC) transfer files, and run transferred binaries through remote Execution. Example execution techniques that rely on authenticated sessions over SMB/RPC are [Scheduled Task/Job](https://attack.mitre.org/techniques/T1053), [Service Execution](https://attack.mitre.org/techniques/T1569/002), and [Windows Management Instrumentation](https://attack.mitre.org/techniques/T1047). Adversaries can also use NTLM hashes to access administrator shares on systems with [Pass the Hash](https://attack.mitre.org/techniques/T1550/002) and certain configuration and patch levels.(Citation: Microsoft Admin Shares)


FirewallInboundConnectionBlocked
ConnectionSuccess
5156
5154
5159
5158
network connection creation
ListeningConnectionCreated
5031
FirewallOutboundConnectionBlocked
ProcessCreated
4688
3
5157
5155
network traffic
4103
command execution
1
command
SMB/Windows Admin Shares
5140
network share
logon session
5145
network share access
4964
logon session creation
4624
4778
LogonSuccess

4. Annexes

< To be corrected or added in future releases >

4.1 List of all techniques used

technique_id	tactic	technique	group
T1059.001	execution	PowerShell	Aquatic Panda, APT41, GALLIUM, APT39, APT19, Thrip, MuddyWater, OilRig, APT29, Deep Panda
T1027	defense-evasion	Obfuscated Files or Information	Aquatic Panda, BackdoorDiplomacy, APT41, GALLIUM, APT39, APT19, MuddyWater, OilRig, APT29
T1588.002	resource-development	Tool	Aquatic Panda, BackdoorDiplomacy, APT41, GALLIUM, APT39, APT19, Thrip, MuddyWater, APT29
T1105	command-and-control	Ingress Tool Transfer	Aquatic Panda, BackdoorDiplomacy, APT41, GALLIUM, APT39, MuddyWater, OilRig, APT29
T1566.001	initial-access	Spearphishing Attachment	APT41, Machete, APT39, APT19, MuddyWater, OilRig, APT29
T1059.003	execution	Windows Command Shell	Aquatic Panda, APT41, Machete, GALLIUM, MuddyWater, OilRig, APT29
T1053.005	execution, persistence, privilege-escalation	Scheduled Task	APT41, Machete, GALLIUM, APT39, MuddyWater, OilRig, APT29
T1071.001	command-and-control	Web Protocols	APT41, APT39, APT19, MuddyWater, OilRig, APT29
T1047	execution	Windows Management Instrumentation	APT41, GALLIUM, MuddyWater, OilRig, APT29, Deep Panda
T1204.002	execution	Malicious File	Machete, APT39, APT19, MuddyWater, OilRig, APT29
T1505.003	persistence	Web Shell	BackdoorDiplomacy, GALLIUM, APT39, OilRig, APT29, Deep Panda
T1036.005	defense-evasion	Match Legitimate Name or Location	BackdoorDiplomacy, APT41, Machete, APT39, MuddyWater, APT29
T1033	discovery	System Owner/User Discovery	APT41, GALLIUM, APT39, APT19, MuddyWater, OilRig
T1003.001	credential-access	LSASS Memory	Aquatic Panda, APT41, GALLIUM, APT39, MuddyWater, OilRig
T1560.001	collection	Archive via Utility	Aquatic Panda, APT41, GALLIUM, APT39, MuddyWater, APT29
T1082	discovery	System Information Discovery	Aquatic Panda, APT19, MuddyWater, OilRig, APT29
T1070.004	defense-evasion	File Deletion	Aquatic Panda, APT41, APT39, OilRig, APT29
T1566.002	initial-access	Spearphishing Link	Machete, APT39, MuddyWater, OilRig, APT29
T1140	defense-evasion	Deobfuscate/Decode Files or Information	APT39, APT19, MuddyWater, OilRig, APT29
T1078	defense-evasion, persistence, privilege-escalation, initial-access	Valid Accounts	APT41, GALLIUM, APT39, OilRig, APT29
T1049	discovery	System Network Connections Discovery	BackdoorDiplomacy, APT41, GALLIUM, MuddyWater, OilRig
T1059.005	execution	Visual Basic	Machete, APT39, MuddyWater, OilRig, APT29
T1204.001	execution	Malicious Link	Machete, APT39, MuddyWater, OilRig, APT29
T1190	initial-access	Exploit Public-Facing Application	BackdoorDiplomacy, APT41, GALLIUM, APT39, APT29
T1547.001	persistence, privilege-escalation	Registry Run Keys / Startup Folder	APT41, APT39, APT19, MuddyWater, APT29
T1016	discovery	System Network Configuration Discovery	APT41, GALLIUM, APT19, MuddyWater, OilRig
T1083	discovery	File and Directory Discovery	APT41, APT39, MuddyWater, APT29
T1018	discovery	Remote System Discovery	GALLIUM, APT39, APT29, Deep Panda
T1021.001	lateral-movement	Remote Desktop Protocol	APT41, APT39, OilRig, APT29
T1021.002	lateral-movement	SMB/Windows Admin Shares	APT41, APT39, APT29, Deep Panda
T1218.011	defense-evasion	Rundll32	APT41, APT19, MuddyWater, APT29
T1555	credential-access	Credentials from Password Stores	APT39, MuddyWater, OilRig, APT29
T1057	discovery	Process Discovery	MuddyWater, OilRig, APT29, Deep Panda
T1133	persistence, initial-access	External Remote Services	APT41, GALLIUM, OilRig, APT29
T1059.006	execution	Python	Machete, APT39, MuddyWater, APT29
T1005	collection	Data from Local System	APT41, GALLIUM, APT39, APT29
T1046	discovery	Network Service Discovery	BackdoorDiplomacy, APT41, APT39, OilRig
T1036.004	defense-evasion	Masquerade Task or Service	BackdoorDiplomacy, APT41, APT29
T1102.002	command-and-control	Bidirectional Communication	APT39, MuddyWater, APT29
T1203	execution	Exploitation for Client Execution	APT41, MuddyWater, APT29

T1574.001	persistence, privilege-escalation, defense-evasion	DLL Search Order Hijacking	Aquatic Panda, BackdoorDiplomacy, APT41
T1562.001	defense-evasion	Disable or Modify Tools	Aquatic Panda, MuddyWater, APT29
T1113	collection	Screen Capture	APT39, MuddyWater, OilRig
T1074.001	collection	Local Data Staging	BackdoorDiplomacy, GALLIUM, APT39
T1059	execution	Command and Scripting Interpreter	APT39, APT19, OilRig
T1087.002	discovery	Domain Account	MuddyWater, OilRig, APT29
T1546.008	privilege-escalation, persistence	Accessibility Features	APT41, APT29, Deep Panda
T1574.002	persistence, privilege-escalation, defense-evasion	DLL Side-Loading	APT41, GALLIUM, APT19
T1553.002	defense-evasion	Code Signing	APT41, GALLIUM, APT29
T1071.004	command-and-control	DNS	APT41, APT39, OilRig
T1555.003	credential-access	Credentials from Web Browsers	MuddyWater, OilRig, APT29
T1041	exfiltration	Exfiltration Over C2 Channel	GALLIUM, APT39, MuddyWater
T1056.001	collection, credential-access	Keylogging	APT41, APT39, OilRig
T1090.002	command-and-control	External Proxy	GALLIUM, APT39, MuddyWater
T1027.002	defense-evasion	Software Packing	GALLIUM, APT39, APT29
T1027.005	defense-evasion	Indicator Removal from Tools	GALLIUM, OilRig, Deep Panda
T1090.001	command-and-control	Internal Proxy	APT39, APT29
T1595.002	reconnaissance	Vulnerability Scanning	Aquatic Panda, APT29
T1021.004	lateral-movement	SSH	APT39, OilRig
T1547.009	persistence, privilege-escalation	Shortcut Modification	APT39, APT29
T1036	defense-evasion	Masquerading	OilRig, APT29
T1069.002	discovery	Domain Groups	OilRig, APT29
T1566.003	initial-access	Spearphishing via Service	OilRig, APT29
T1218.005	defense-evasion	Mshta	MuddyWater, APT29
T1548.002	privilege-escalation, defense-evasion	Bypass User Account Control	MuddyWater, APT29
T1552.001	credential-access	Credentials In Files	MuddyWater, OilRig
T1003.005	credential-access	Cached Domain Credentials	MuddyWater, OilRig
T1003.004	credential-access	LSA Secrets	MuddyWater, OilRig
T1583.006	resource-development	Web Services	MuddyWater, APT29
T1219	command-and-control	Remote Access Software	Thrip, MuddyWater
T1048.003	exfiltration	Exfiltration Over Unencrypted Non-C2 Protocol	Thrip, OilRig
T1189	initial-access	Drive-by Compromise	Machete, APT19
T1564.003	defense-evasion	Hidden Window	APT19, Deep Panda
T1132.001	command-and-control	Standard Encoding	APT19, MuddyWater
T1012	discovery	Query Registry	APT39, OilRig
T1218.010	defense-evasion	Regsvr32	APT19, Deep Panda
T1110	credential-access	Brute Force	APT39, OilRig
T1104	command-and-control	Multi-Stage Channels	APT41, MuddyWater
T1518.001	discovery	Security Software Discovery	Aquatic Panda, MuddyWater
T1197	defense-evasion, persistence	BITS Jobs	APT41, APT39
T1135	discovery	Network Share Discovery	APT41, APT39
T1112	defense-evasion	Modify Registry	APT41, APT19
T1136.001	persistence	Local Account	APT41, APT39
T1120	discovery	Peripheral Device Discovery	BackdoorDiplomacy, OilRig
T1588.001	resource-development	Malware	Aquatic Panda, BackdoorDiplomacy
T1569.002	execution	Service Execution	APT41, APT39
T1095	command-and-control	Non-Application Layer Protocol	BackdoorDiplomacy, APT29
T1543.003	persistence, privilege-escalation	Windows Service	APT41, APT19
T1195.002	initial-access	Compromise Software Supply Chain	APT41, APT29
T1218.001	defense-evasion	Compiled HTML File	APT41, OilRig

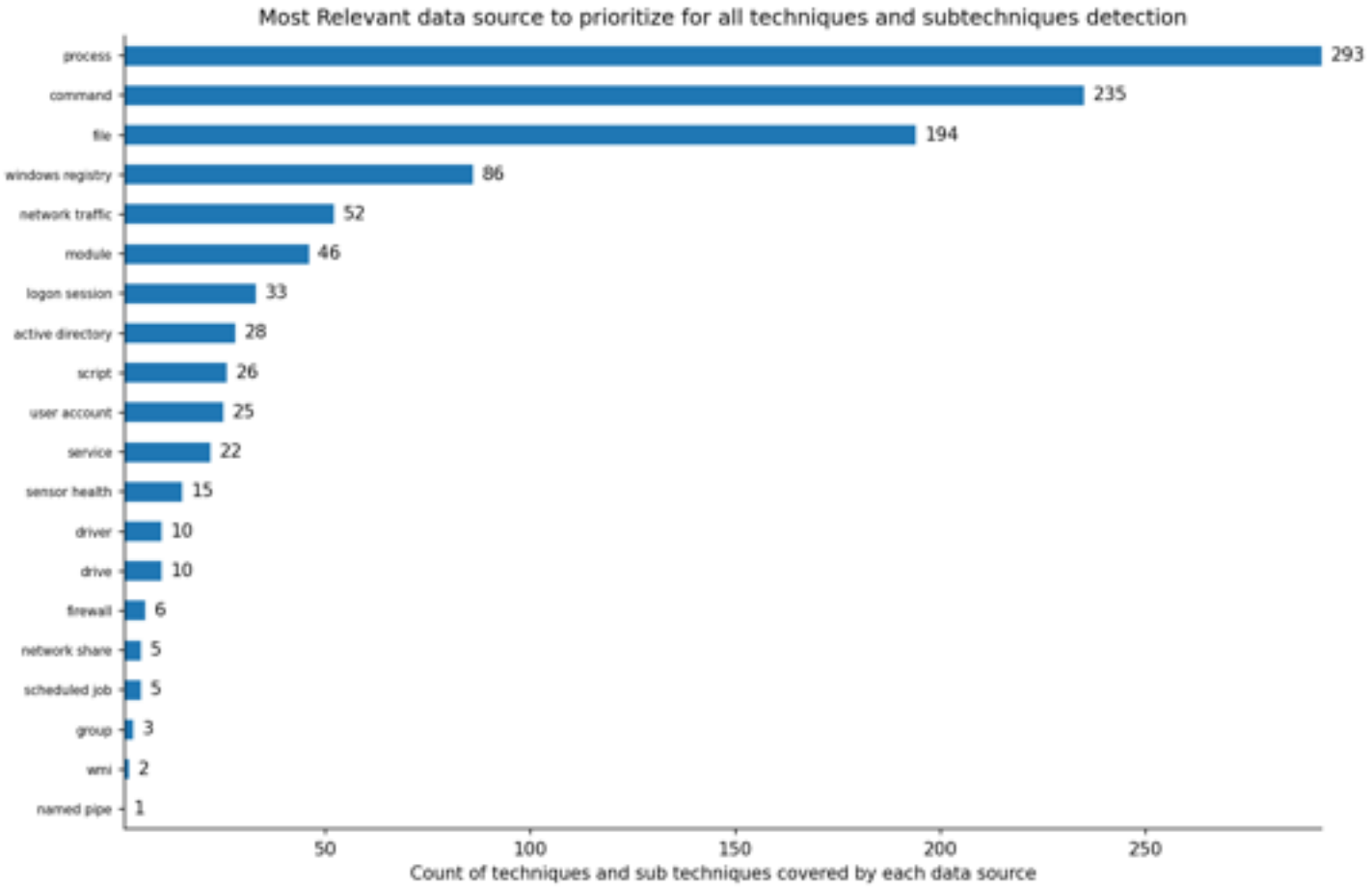
T1008	command-and-control	Fallback Channels	APT41, OilRig
T1007	discovery	System Service Discovery	Aquatic Panda, OilRig
T1048.002	exfiltration	Exfiltration Over Asymmetric Encrypted Non-C2 Protocol	APT29
T1001.002	command-and-control	Steganography	APT29
T1583.004	resource-development	Server	GALLIUM
T1090.004	command-and-control	Domain Fronting	APT29
T1090.003	command-and-control	Multi-hop Proxy	APT29
T1074.002	collection	Remote Data Staging	APT29
T1078.004	defense-evasion, persistence, privilege-escalation, initial-access	Cloud Accounts	APT29
T1078.003	defense-evasion, persistence, privilege-escalation, initial-access	Local Accounts	APT29
T1573	command-and-control	Encrypted Channel	APT29
T1586.002	resource-development	Email Accounts	APT29
T1583.001	resource-development	Domains	APT29
T1584.001	resource-development	Domains	APT29
T1568	command-and-control	Dynamic Resolution	APT29
T1587.001	resource-development	Malware	APT29
T1587.003	resource-development	Digital Certificates	APT29
T1589.001	reconnaissance	Credentials	APT29
T1606.001	credential-access	Web Cookies	APT29
T1606.002	credential-access	SAML Tokens	APT29
T1484.002	defense-evasion, privilege-escalation	Domain Trust Modification	APT29
T1553.005	defense-evasion	Mark-of-the-Web Bypass	APT29
T1016.001	discovery	Internet Connection Discovery	APT29
T1213.003	collection	Code Repositories	APT29
T1078.002	defense-evasion, persistence, privilege-escalation, initial-access	Domain Accounts	APT29
T1003.006	credential-access	DCSync	APT29
T1087.004	discovery	Cloud Account	APT29
T1562.004	defense-evasion	Disable or Modify System Firewall	APT29
T1069	discovery	Permission Groups Discovery	APT29
T1070	defense-evasion	Indicator Removal on Host	APT29
T1087	discovery	Account Discovery	APT29
T1213	collection	Data from Information Repositories	APT29
T1199	initial-access	Trusted Relationship	APT29
T1482	discovery	Domain Trust Discovery	APT29
T1539	credential-access	Steal Web Session Cookie	APT29
T1098.001	persistence	Additional Cloud Credentials	APT29
T1098.002	persistence	Additional Email Delegate Permissions	APT29
T1098.003	persistence	Additional Cloud Roles	APT29
T1546.003	privilege-escalation, persistence	Windows Management Instrumentation Event Subscription	APT29
T1136.003	persistence	Cloud Account	APT29
T1550	defense-evasion, lateral-movement	Use Alternate Authentication Material	APT29
T1550.003	defense-evasion, lateral-movement	Pass the Ticket	APT29
T1550.001	defense-evasion, lateral-movement	Application Access Token	APT29
T1550.004	defense-evasion, lateral-movement	Web Session Cookie	APT29
T1070.006	defense-evasion	Timestomp	APT29
T1552.004	credential-access	Private Keys	APT29
T1027.001	defense-evasion	Binary Padding	APT29
T1021.006	lateral-movement	Windows Remote Management	APT29
T1110.003	credential-access	Password Spraying	APT29
T1558.003	credential-access	Kerberoasting	APT29

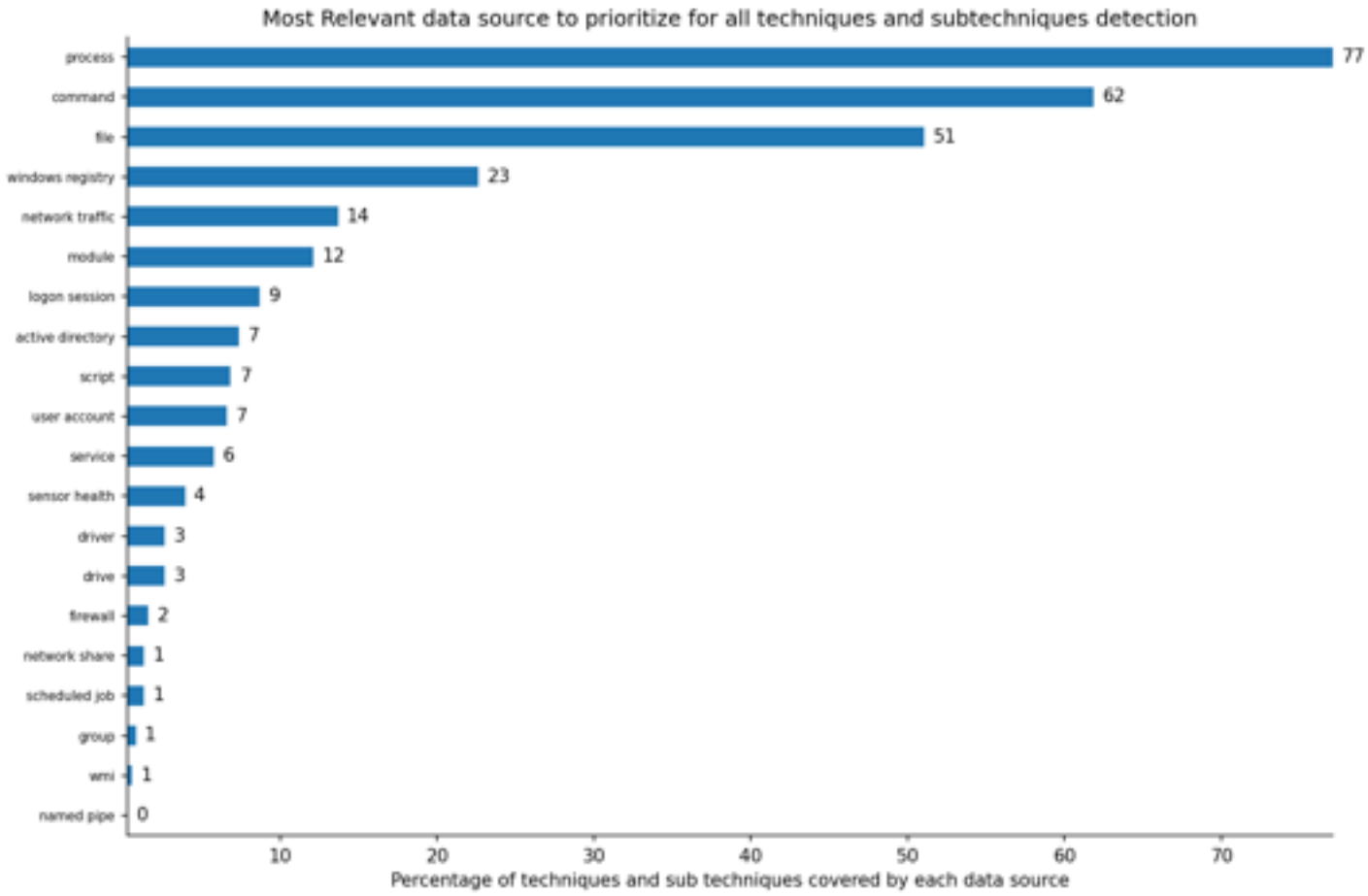
T1098.005	persistence	Device Registration	APT29
T1114.002	collection	Remote Email Collection	APT29
T1562.002	defense-evasion	Disable Windows Event Logging	APT29
T1027.006	defense-evasion	HTML Smuggling	APT29
T1119	collection	Automated Collection	OilRig
T1621	credential-access	Multi-Factor Authentication Request Generation	APT29
T1496	impact	Resource Hijacking	APT41
T1589.002	reconnaissance	Email Addresses	MuddyWater
T1568.002	command-and-control	Domain Generation Algorithms	APT41
T1059.004	execution	Unix Shell	APT41
T1110.002	credential-access	Password Cracking	APT41
T1070.003	defense-evasion	Clear Command History	APT41
T1070.001	defense-evasion	Clear Windows Event Logs	APT41
T1003	credential-access	OS Credential Dumping	APT39
T1056	collection, credential-access	Input Capture	APT39
T1542.003	persistence, defense-evasion	Bootkit	APT41
T1115	collection	Clipboard Data	APT39
T1546.010	privilege-escalation, persistence	AppInit DLLs	APT39
T1027.004	defense-evasion	Compile After Delivery	MuddyWater
T1486	impact	Data Encrypted for Impact	APT41
T1553.006	defense-evasion	Code Signing Policy Modification	APT39
T1090	command-and-control	Proxy	APT41
T1055	defense-evasion, privilege-escalation	Process Injection	APT41
T1136.002	persistence	Domain Account	GALLIUM
T1550.002	defense-evasion, lateral-movement	Pass the Hash	GALLIUM
T1014	defense-evasion	Rootkit	APT41
T1036.003	defense-evasion	Rename System Utilities	GALLIUM
T1003.002	credential-access	Security Account Manager	GALLIUM
T1570	lateral-movement	Lateral Tool Transfer	GALLIUM
T1059.007	execution	JavaScript	MuddyWater
T1574.006	persistence, privilege-escalation, defense-evasion	Dynamic Linker Hijacking	APT41
T0869	command-and-control-ics	Standard Application Layer Protocol	OilRig
T1572	command-and-control	Protocol Tunneling	OilRig
T0859	persistence-ics, lateral-movement-ics	Valid Accounts	OilRig
T0817	initial-access-ics	Drive-by Compromise	OilRig
T0865	initial-access-ics	Spearphishing Attachment	OilRig
T0853	execution-ics	Scripting	OilRig
T1218.007	defense-evasion	Msiexec	Machete
T1201	discovery	Password Policy Discovery	OilRig
T1137.004	persistence	Outlook Home Page	OilRig
T1087.001	discovery	Local Account	OilRig
T1497.001	defense-evasion, discovery	System Checks	OilRig
T1069.001	discovery	Local Groups	OilRig
T1573.002	command-and-control	Asymmetric Cryptography	OilRig
T1102.001	command-and-control	Dead Drop Resolver	APT41
T1555.004	credential-access	Windows Credential Manager	OilRig
T1055.001	defense-evasion, privilege-escalation	Dynamic-link Library Injection	BackdoorDiplomacy
T1518	discovery	Software Discovery	MuddyWater
T1137.001	persistence	Office Template Macros	MuddyWater
T1218.003	defense-evasion	CMSTP	MuddyWater

T1480.001	defense-evasion	Environmental Keying	APT41
T1027.003	defense-evasion	Steganography	MuddyWater
T1559.001	execution	Component Object Model	MuddyWater
T1559.002	execution	Dynamic Data Exchange	MuddyWater
T1071.002	command-and-control	File Transfer Protocols	APT41
T1068	privilege-escalation	Exploitation for Privilege Escalation	APT29

4.2 Data sources reference for covering all mitre technique

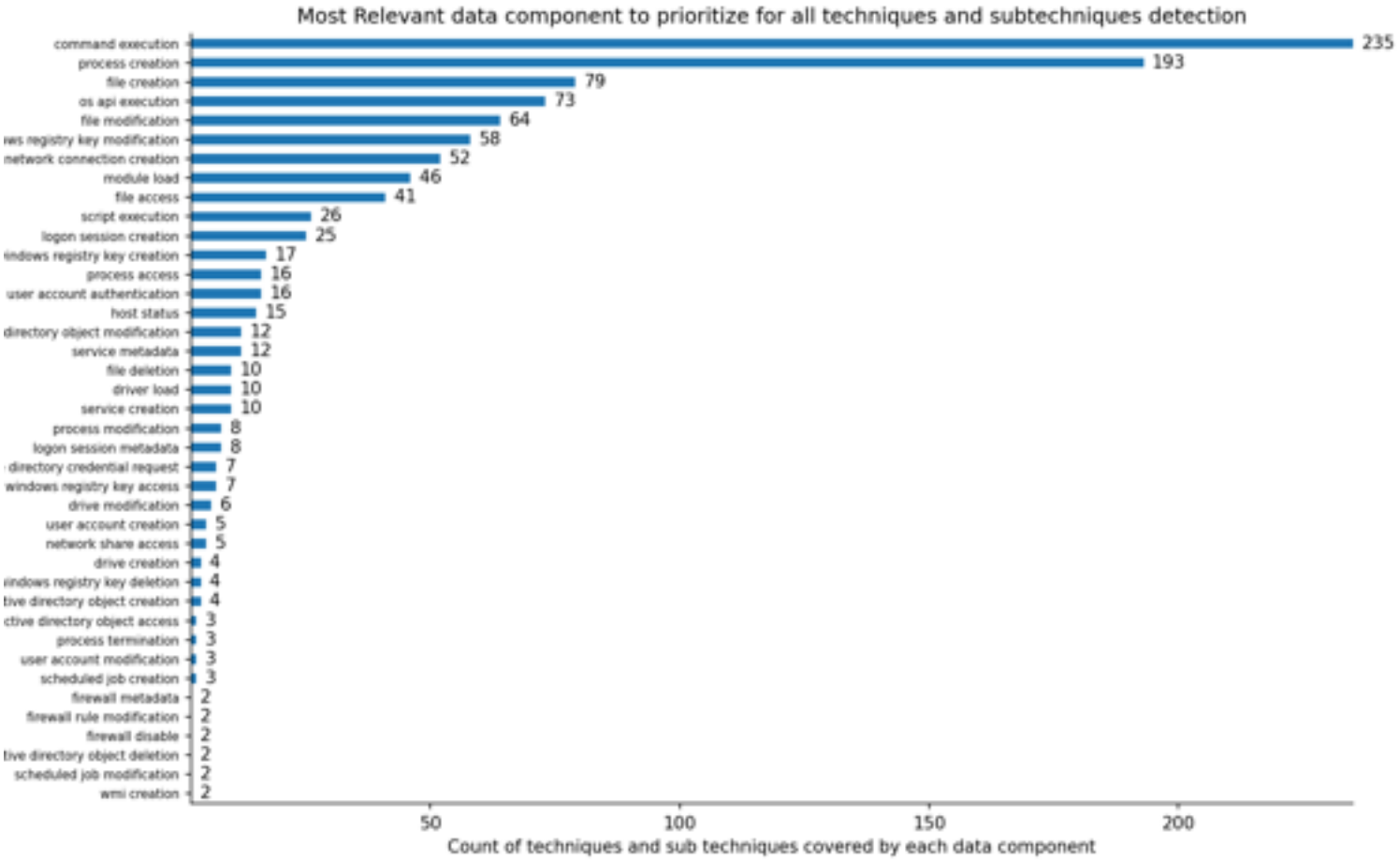
< To be corrected or added in future releases >

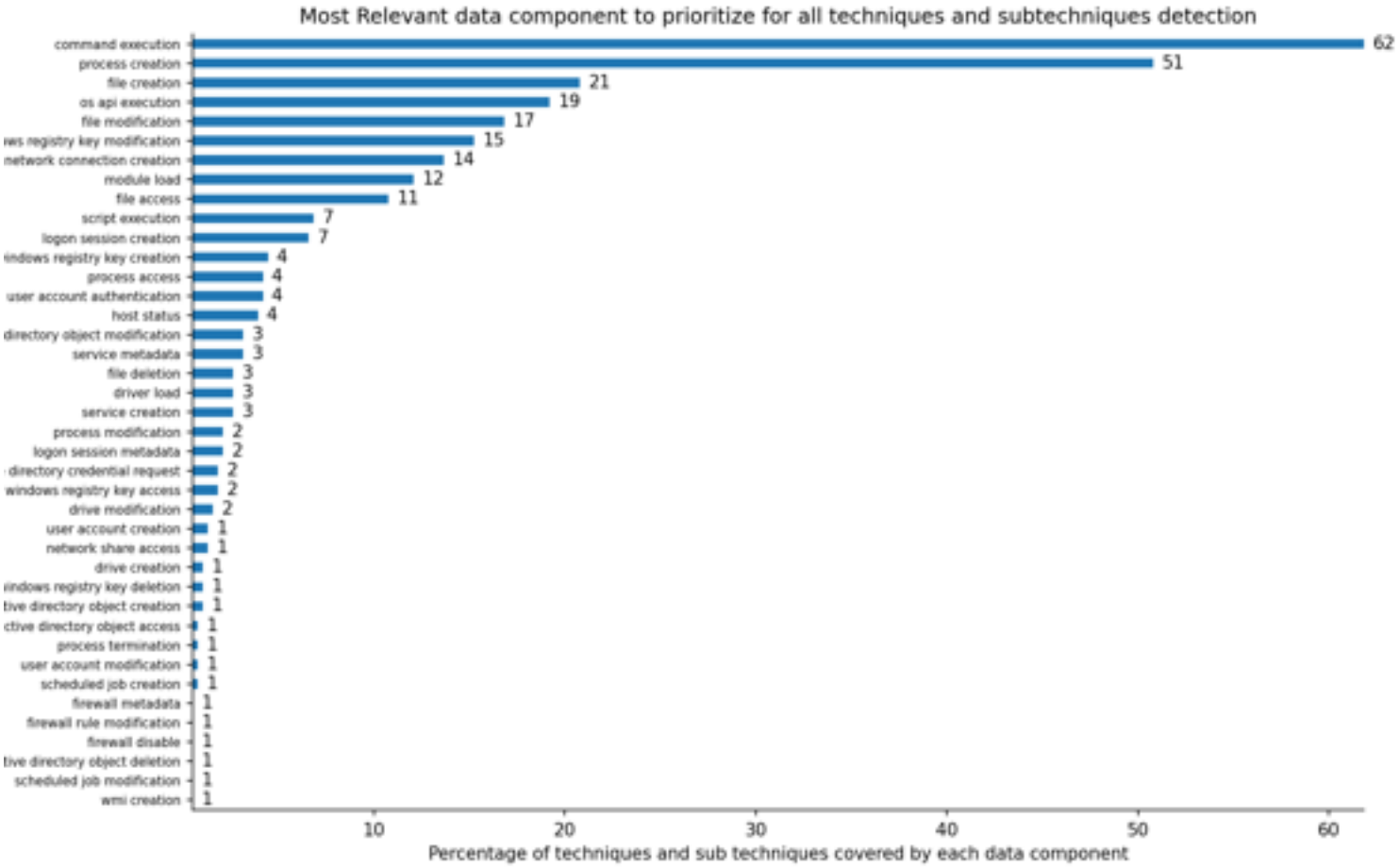




4.3 Data component reference for covering all mitre technique

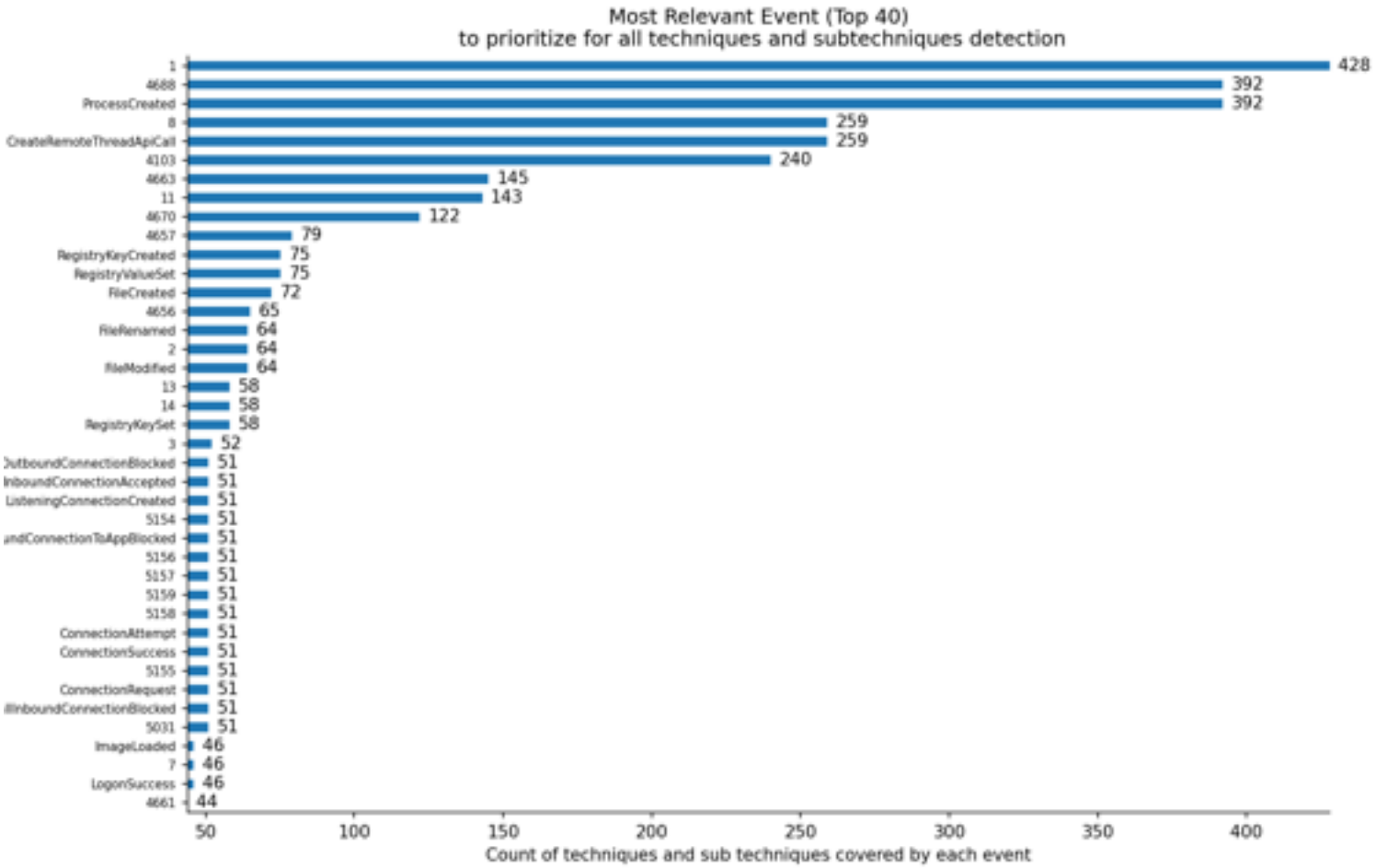
< To be corrected or added in future releases >

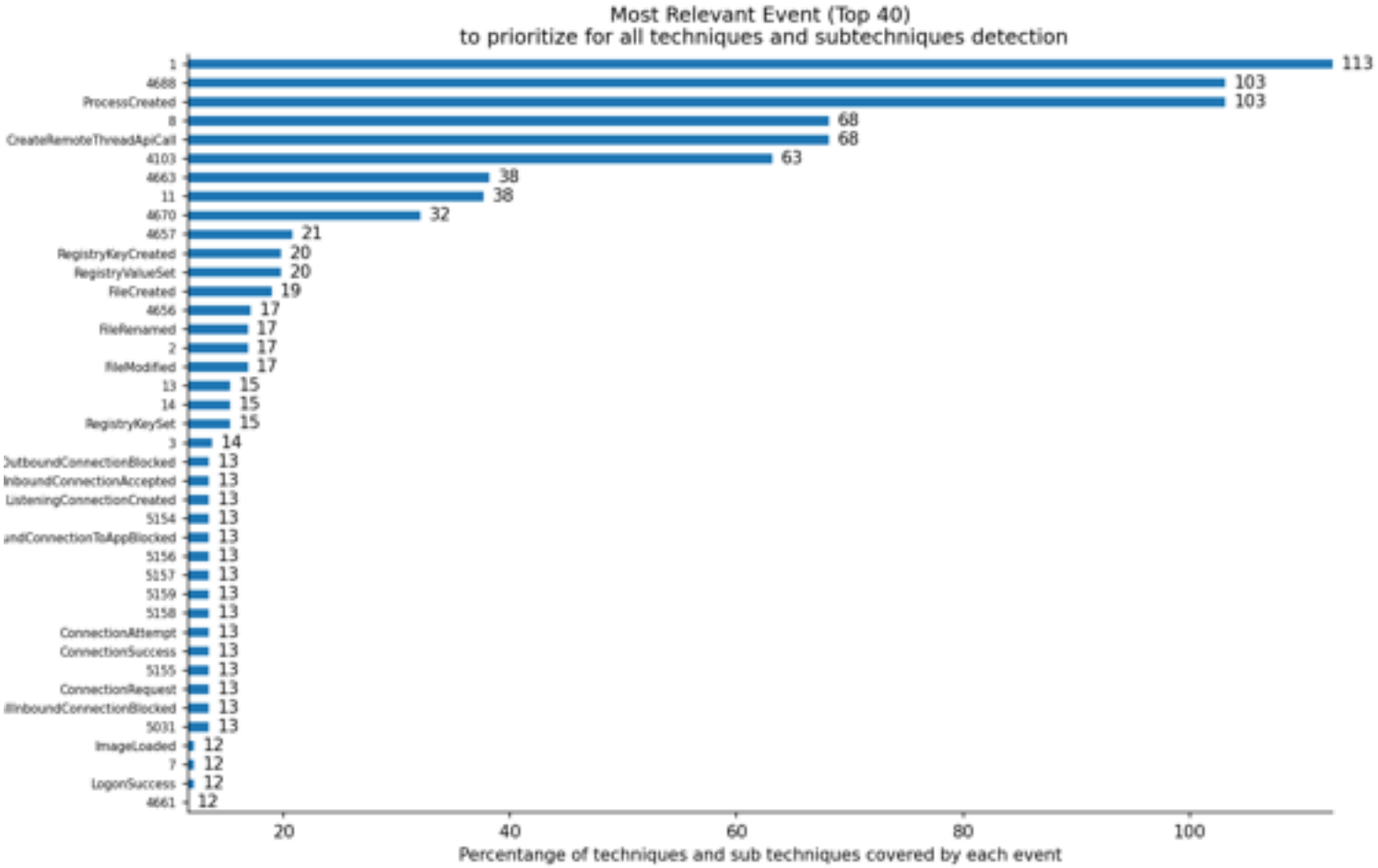

Most Relevant data component to prioritize for all techniques and subtechniques detection
command execution 235
process creation 193
file creation 79
os api execution 73
file modification 64
ws registry key modification 58
network connection creation 52
module load 46
file access 41
script execution 26
logon session creation 25
indows registry key creation 17
process access 16
user account authentication 16
host status 15
directory object modification 12
service metadata 12
file deletion 10
driver load 10
service creation 10
process modification 8
logon session metadata 8
directory credential request 7
windows registry key access 7
drive modification 6
user account creation 5
network share access 5
drive creation 4
indows registry key deletion 4
tive directory object creation 4
ctive directory object access 3
process termination 3
user account modification 3
scheduled job creation 3
firewall metadata 2
firewall rule modification 2
firewall disable 2
tive directory object deletion 2
scheduled job modification 2
wmi creation 2
50
100
150
200
Count of techniques and sub techniques covered by each data component



4.4 Event reference for covering all mitre technique

< To be corrected or added in future releases >


Most Relevant Event (Top 40)
to prioritize for all techniques and subtechniques detection
1
4688
ProcessCreated
8
CreateRemoteThreadApiCall
4103
4663
11
4670
4657
RegistryKeyCreated
RegistryValueSet
FileCreated
4656
FileRenamed
2
fileModified
13
14
RegistryKeySet
3
OutboundConnectionBlocked
InboundConnectionAccepted
ListeningConnectionCreated
5154
andConnectionToAppBlocked
5156
5157
5159
5158
ConnectionAttempt
ConnectionSuccess
5155
ConnectionRequest
dInboundConnectionBlocked
5031
ImageLoaded
7
LogonSuccess
4661
428
392
392
259
259
240
145
143
122
79
75
75
72
65
64
64
64
58
58
58
52
51
51
51
51
51
51
51
51
51
51
51
51
51
51
51
46
46
46
44
50
100
150
200
250
300
350
400
Count of techniques and sub techniques covered by each event


Most Relevant Event (Top 40)
to prioritize for all techniques and subtechniques detection
1
4688
ProcessCreated
8
CreateRemoteThreadApiCall
4103
4663
11
4670
4657
RegistryKeyCreated
RegistryValueSet
FileCreated
4656
FileRenamed
2
fileModified
13
14
RegistryKeySet
3
utboundConnectionBlocked
nboundConnectionAccepted
ListeningConnectionCreated
5154
ndConnectionToAppBlocked
5156
5157
5159
5158
ConnectionAttempt
ConnectionSuccess
5155
ConnectionRequest
lInboundConnectionBlocked
5031
ImageLoaded
7
LogonSuccess
4661
113
103
103
68
68
63
38
38
32
21
20
20
19
17
17
17
17
15
15
15
14
13
13
13
13
13
13
13
13
13
13
13
13
13
13
13
12
12
12
12
20
40
60
80
100
Percentange of techniques and sub techniques covered by each event