Must Have SOC Analysts customized cookbook

Leverage automation and threat intel data analysis for prioritizing detection

Report customized for energy sector

This report aims at providing statical analysis of TTPs (Tactics, Techniques and Procedures) used by threat actors targetting energy 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 energy 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 Tonto Team

Alias : Tonto Team, Earth Akhlut, BRONZE HUNTLEY, CactusPete, Karma Panda

Tonto Team (https://attack.mitre.org/groups/G0131) is a suspected Chinese state-sponsored cyber espionage threat group that has primarily targeted South Korea, Japan, Taiwan, and the United States since at least 2009; by 2020 they expanded operations to include other Asian as well as Eastern European countries. Tonto Team (https://attack.mitre.org/groups/G0131) has targeted government, military, energy, mining, financial, education, healthcare, and technology organizations, including through the Heartbeat Campaign (2009-2012) and Operation Bitter Biscuit (2017).
Citation: Kaspersky CactusPete Aug 2020
Citation: ESET Exchange Mar 2021
Citation: FireEye Chinese Espionage October 2019
Citation: ARS Technica China Hack SK April 2017
Citation: Trend Micro HeartBeat Campaign January 2013
Citation: Talos Bisonal 10 Years March 2020

1.2 Operation Wocao

Alias : Operation Wocao

Operation Wocao (https://attack.mitre.org/groups/G0116) described activities carried out by a China-based cyber espionage adversary. Operation Wocao (https://attack.mitre.org/groups/G0116) targeted entities within the government, managed service providers, energy, health care, and technology sectors across several countries, including China, France, Germany, the United Kingdom, and the United States. Operation Wocao (https://attack.mitre.org/groups/G0116) used similar TTPs and tools to APT20, suggesting a possible overlap.
Citation: FoxIT Wocao December 2019

1.3 Sharpshooter

Alias : Sharpshooter

Operation Sharpshooter (https://attack.mitre.org/groups/G0104) is the name of a cyber espionage campaign discovered in October 2018 targeting nuclear, defense, energy, and financial companies. Though overlaps between this adversary and Lazarus Group (https://attack.mitre.org/groups/G0032) have been noted, definitive links have not been established.
Citation: McAfee Sharpshooter December 2018

1.4 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

1.5 APT33

Alias : APT33, HOLMIUM, Elfin

APT33 (https://attack.mitre.org/groups/G0064) is a suspected Iranian threat group that has carried out operations since at least 2013. The group has targeted organizations across multiple industries in the United States, Saudi Arabia, and South Korea, with a particular interest in the aviation and energy sectors.
Citation: FireEye APT33 Sept 2017
Citation: FireEye APT33 Webinar Sept 2017

1.6 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

1.7 menuPass

Alias : menuPass, Cicada, POTASSIUM, Stone Panda, APT10, Red Apollo, CVNX, HOGFISH

menuPass (https://attack.mitre.org/groups/G0045) is a threat group that has been active since at least 2006. Individual members of menuPass (https://attack.mitre.org/groups/G0045) are known to have acted in association with the Chinese Ministry of State Security's (MSS) Tianjin State Security Bureau and worked for the Huaying Haitai Science and Technology Development Company.
Citation: DOJ APT10 Dec 2018
Citation: District Court of NY APT10 Indictment December 2018

menuPass (https://attack.mitre.org/groups/G0045) has targeted healthcare, defense, aerospace, finance, maritime, biotechnology, energy, and government sectors globally, with an emphasis on Japanese organizations. In 2016 and 2017, the group is known to have targeted managed IT service providers (MSPs), manufacturing and mining companies, and a university.
Citation: Palo Alto menuPass Feb 2017
Citation: Crowdstrike CrowdCast Oct 2013
Citation: FireEye Poison Ivy
Citation: PWC Cloud Hopper April 2017
Citation: FireEye APT10 April 2017
Citation: DOJ APT10 Dec 2018
Citation: District Court of NY APT10 Indictment December 2018

1.8 Threat Group-3390

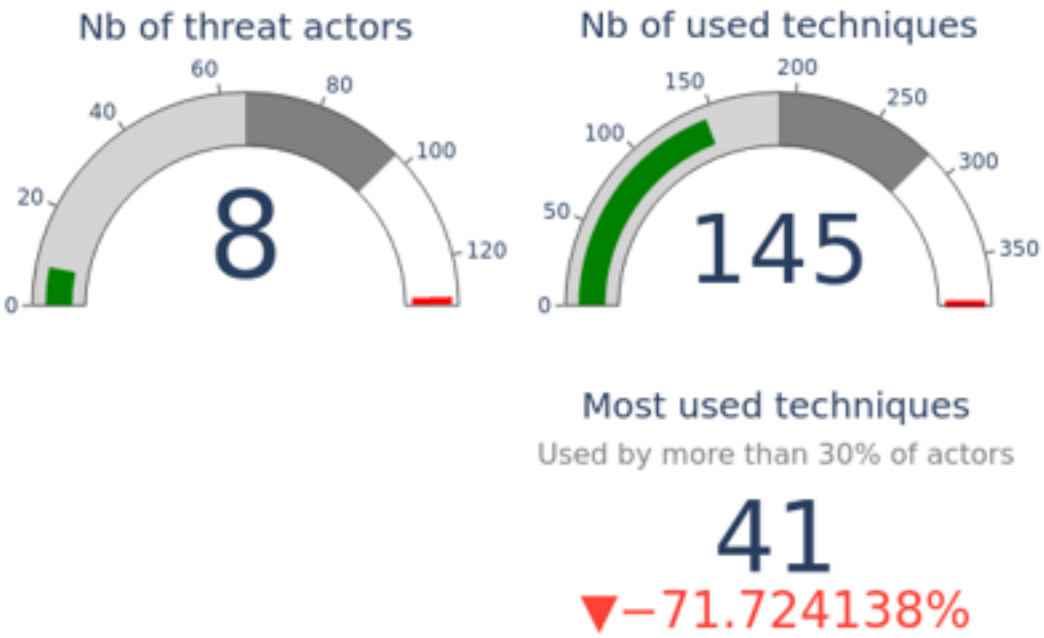
Alias : Threat Group-3390, Earth Smilodon, TG-3390, Emissary Panda, BRONZE UNION, APT27, Iron Tiger, LuckyMouse

[Threat Group-3390](https://attack.mitre.org/groups/G0027) is a Chinese threat group that has extensively used strategic Web compromises to target victims.
Citation: Dell TG-3390

The group has been active since at least 2010 and has targeted organizations in the aerospace, government, defense, technology, energy, manufacturing and gambling/betting sectors.
Citation: SecureWorks BRONZE UNION June 2017
Citation: Securelist LuckyMouse June 2018
Citation: Trend Micro DRBControl February 2020

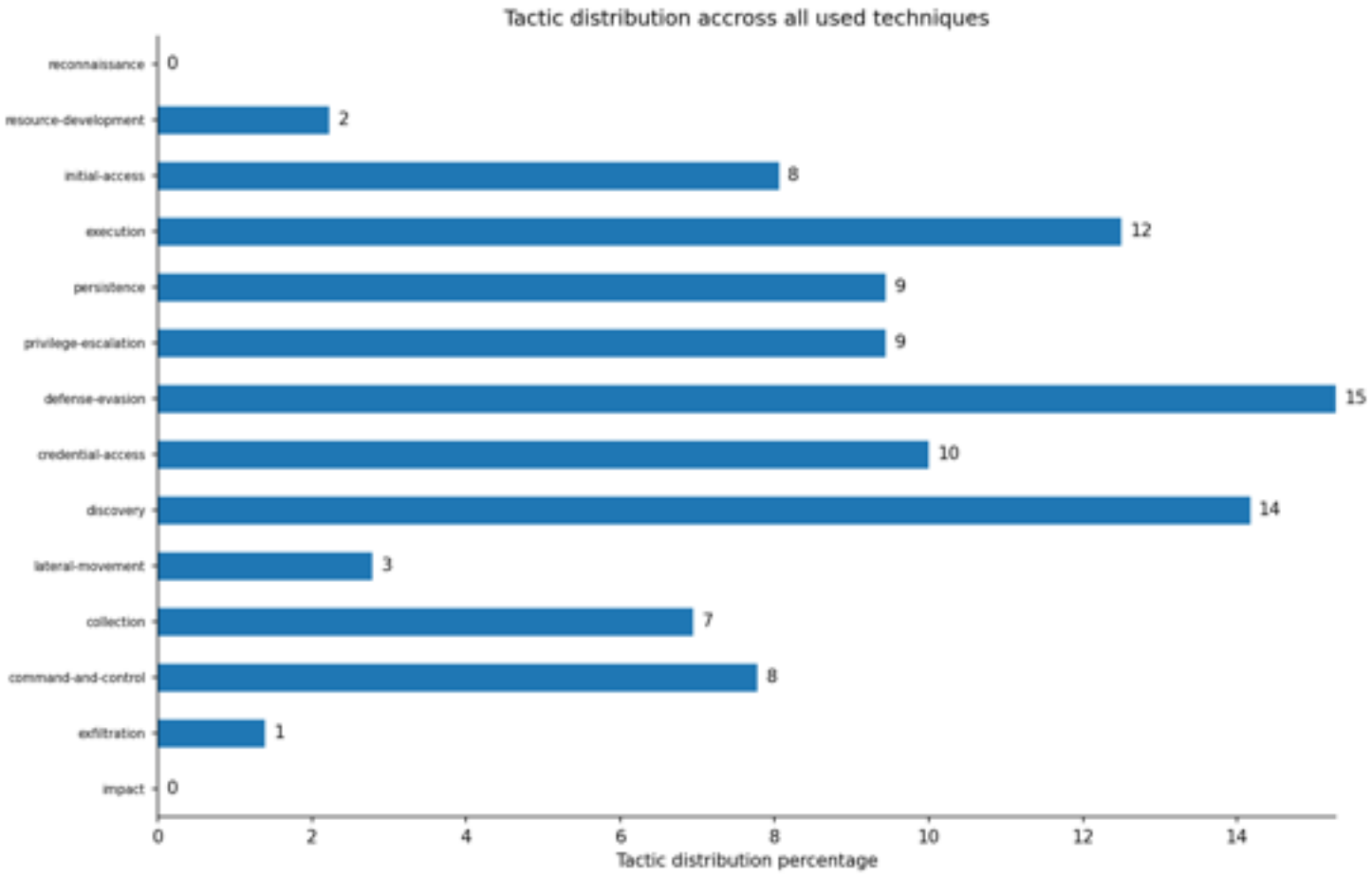
2. What TTPs to prioritize for detection ?

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.



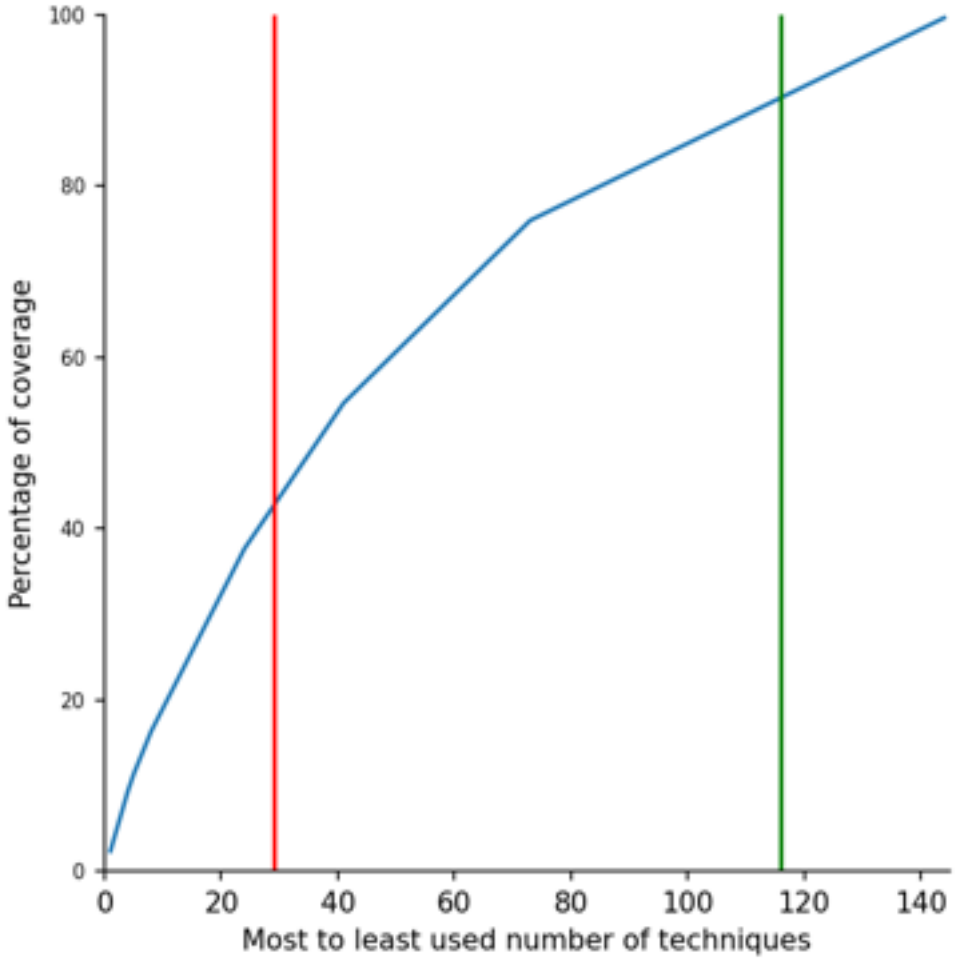
2.1 Tactics distribution

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.



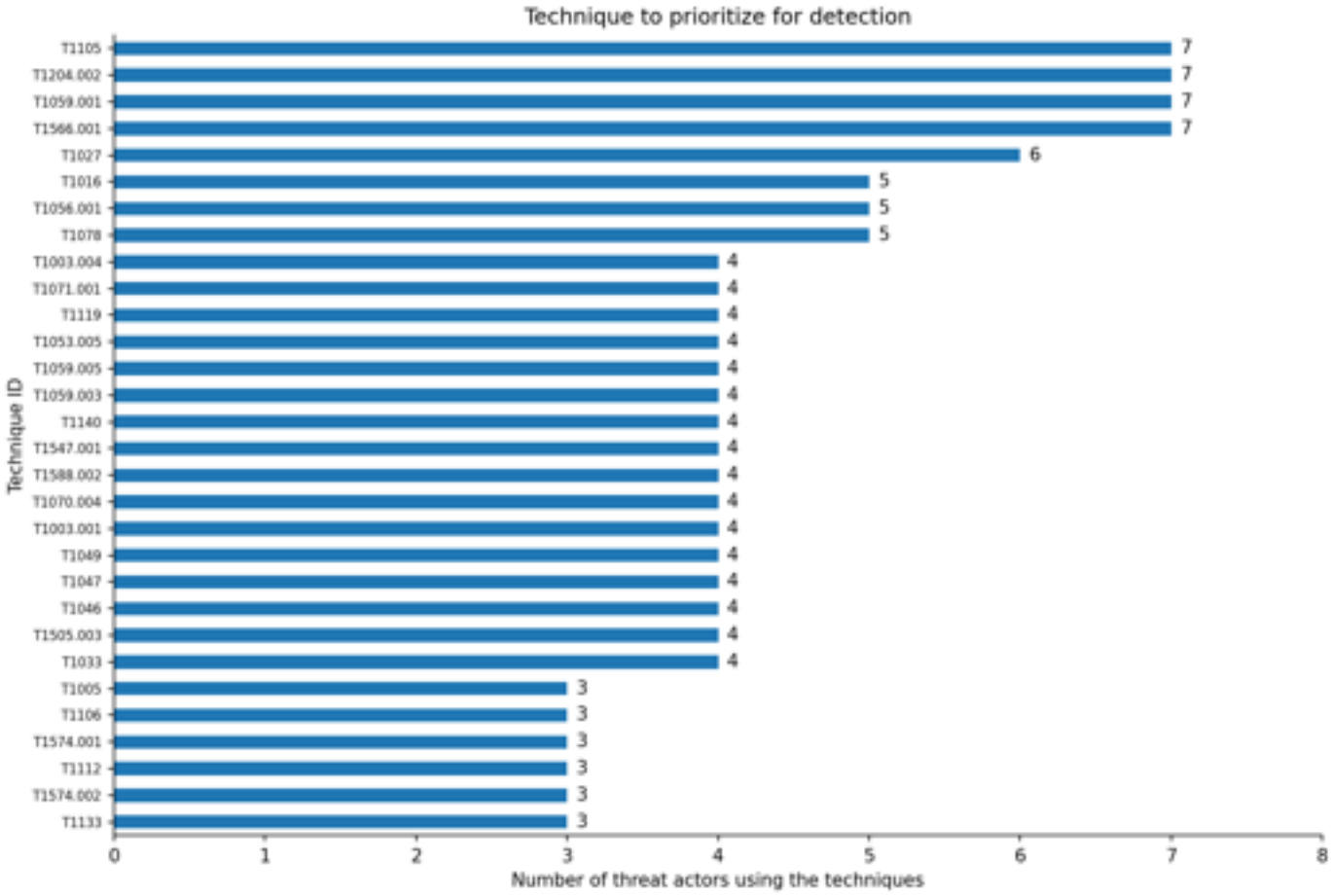
2.2 Technique distribution

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.



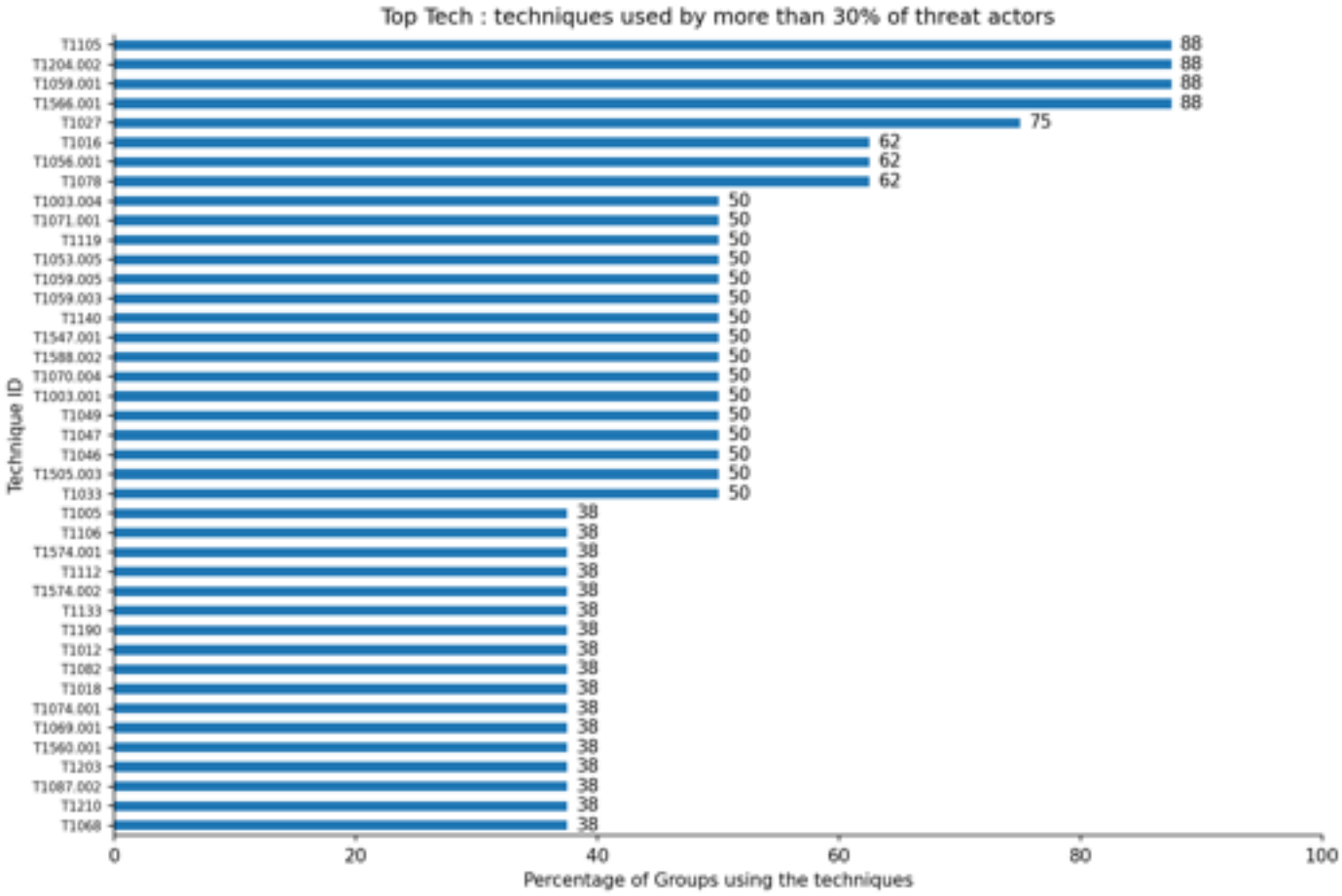
2.3 Top 30 most used techniques

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.



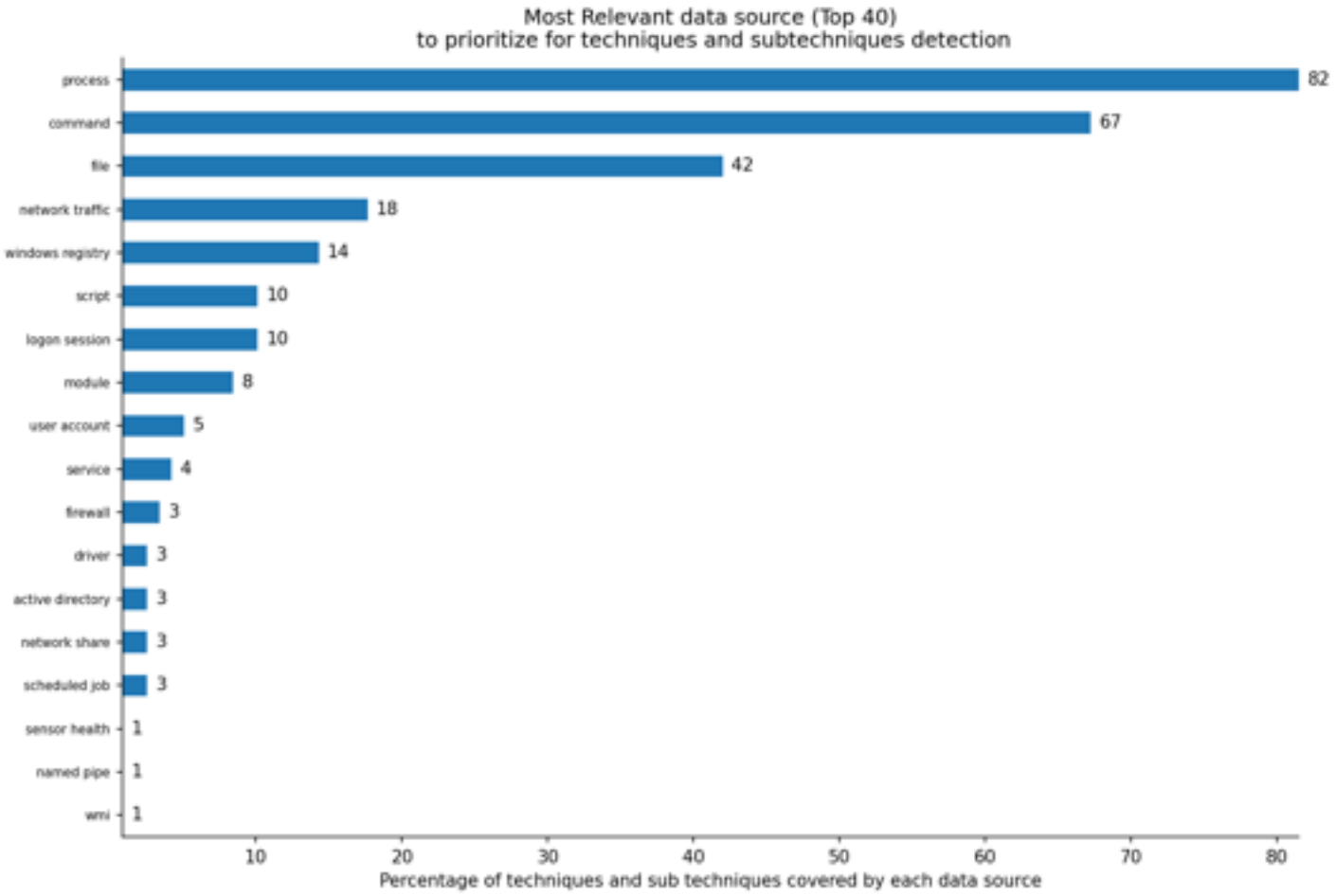
2.4 The Must be covered techniques

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.



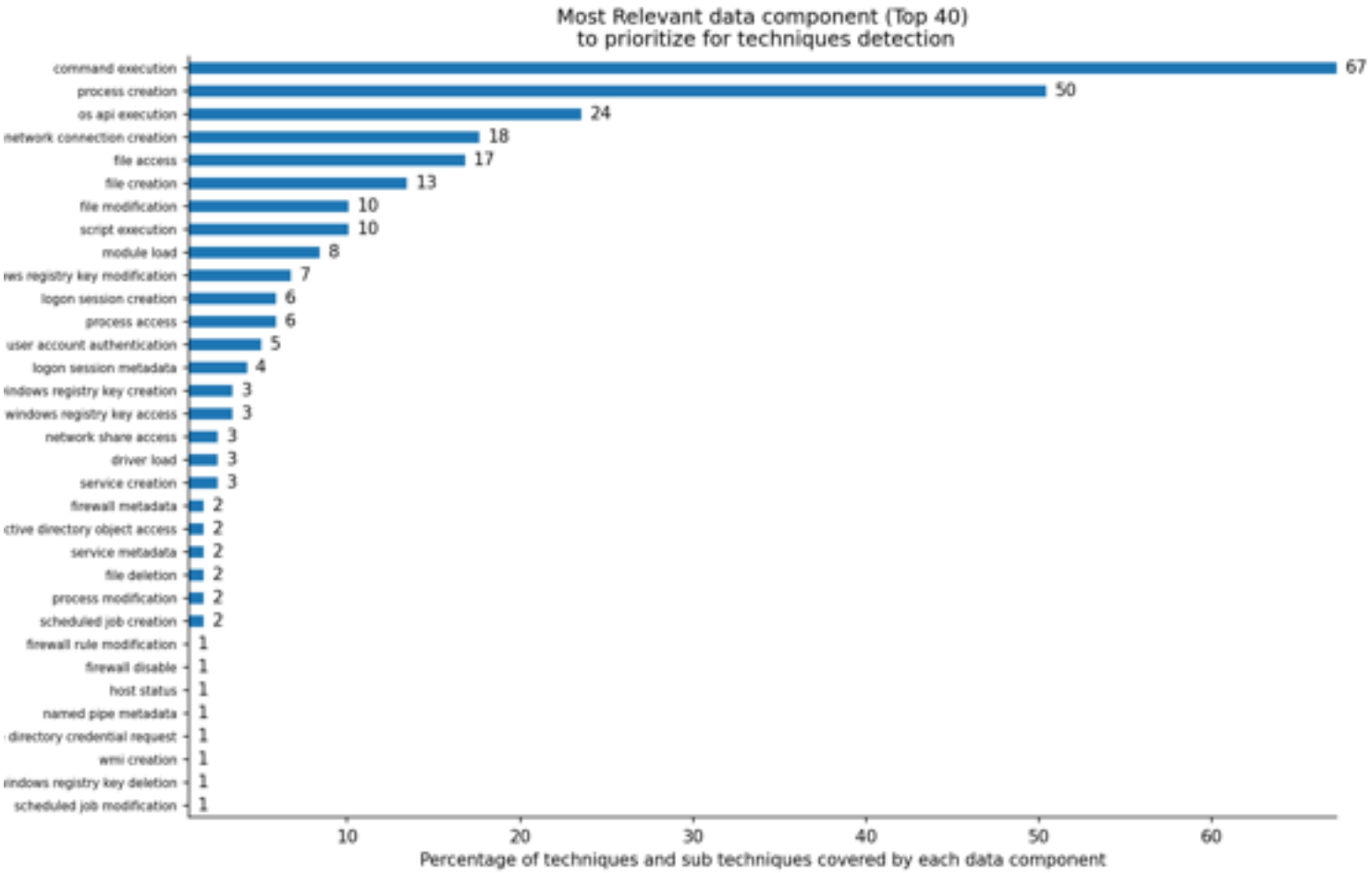
2.5 Top data source to collect for detections

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.



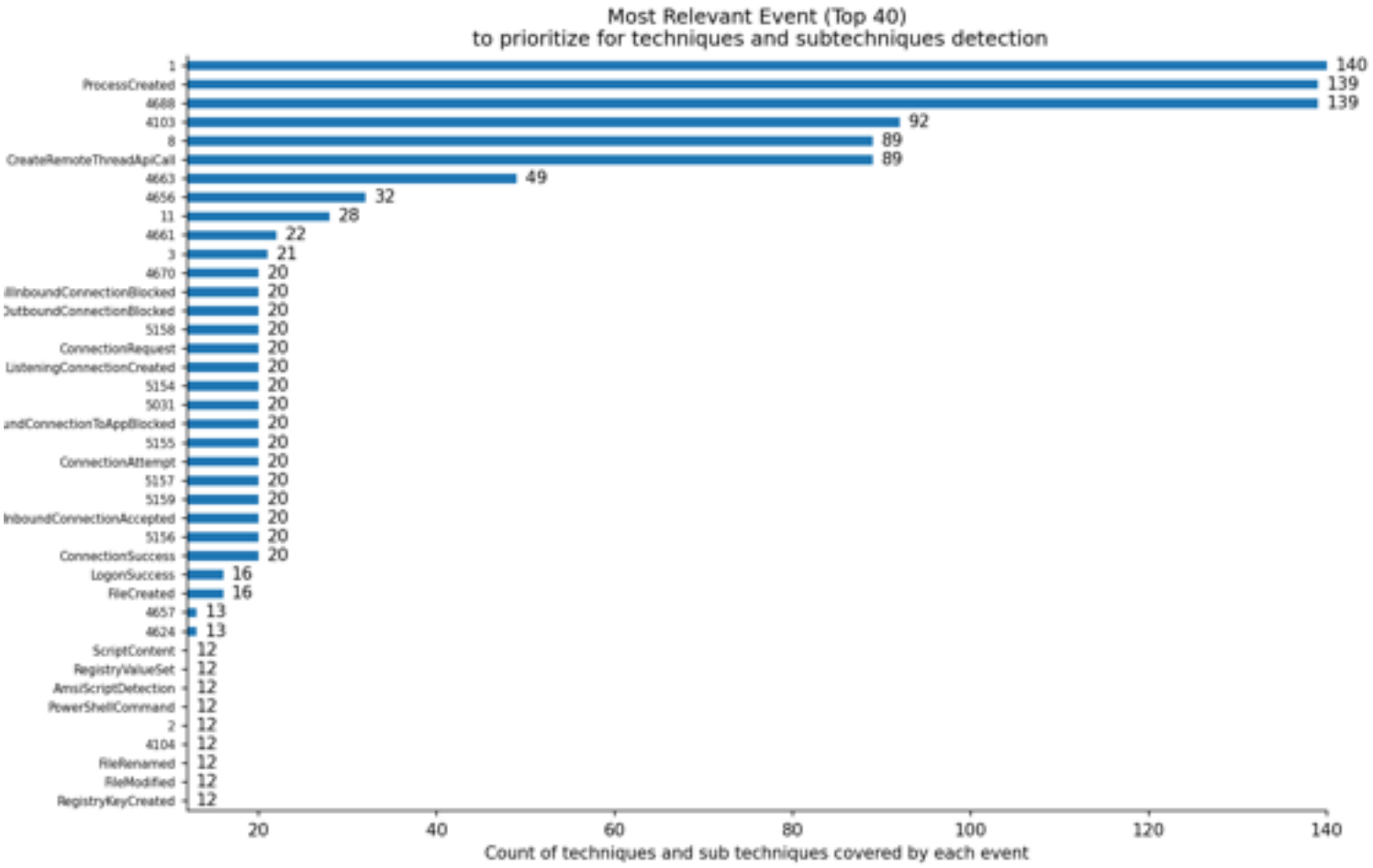
2.6 Top data component to collect for detections

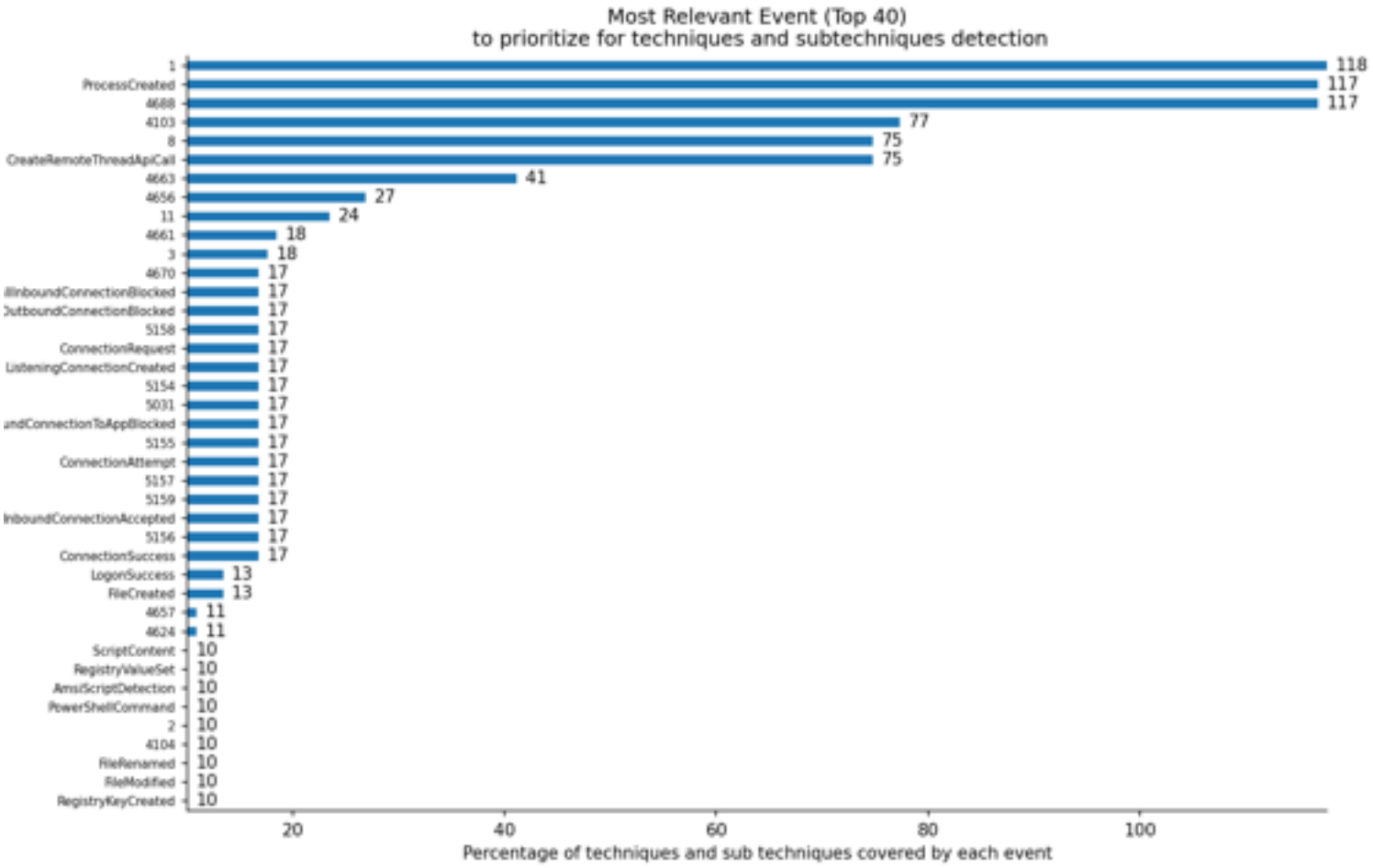
The following graph gives the top 40 data source to collect in order to be able to detect the techniques used by threat actors.
Please see annexes for reference.



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
ProcessCreated
4688
4103
8
CreateRemoteThreadApiCall
4663
4656
11
4661
3
4670
llInboundConnectionBlocked
OutboundConnectionBlocked
5158
ConnectionRequest
ListeningConnectionCreated
5154
5031
ndConnectionToAppBlocked
5155
ConnectionAttempt
5157
5159
InboundConnectionAccepted
5156
ConnectionSuccess
LogonSuccess
FileCreated
4657
4624
ScriptContent
RegistryValueSet
AmsiScriptDetection
PowerShellCommand
2
4104
FileRenamed
FileModified
RegistryKeyCreated
140
139
139
92
89
89
49
32
28
22
21
20
20
20
20
20
20
20
20
20
20
20
20
20
20
20
16
16
13
13
12
12
12
12
12
12
12
12
12
20
40
60
80
100
120
140
Count of techniques and sub techniques covered by each event


Most Relevant Event (Top 40)
to prioritize for techniques and subtechniques detection
1
ProcessCreated
4688
4103
8
CreateRemoteThreadApiCall
4663
4656
11
4661
3
4670
InboundConnectionBlocked
OutboundConnectionBlocked
5158
ConnectionRequest
ListeningConnectionCreated
5154
5031
andConnectionToAppBlocked
5155
ConnectionAttempt
5157
5159
InboundConnectionAccepted
5156
ConnectionSuccess
LogonSuccess
FileCreated
4657
4624
ScriptContent
RegistryValueSet
AmsiScriptDetection
PowerShellCommand
2
4104
FileRenamed
FileModified
RegistryKeyCreated
118
117
117
77
75
75
41
27
24
18
18
17
17
17
17
17
17
17
17
17
17
17
17
17
17
17
13
13
11
11
10
10
10
10
10
10
10
10
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 T1105

Used by group : Tonto Team, Operation Wocao, Sharpshooter, APT33, OilRig, menuPass, Threat Group-3390

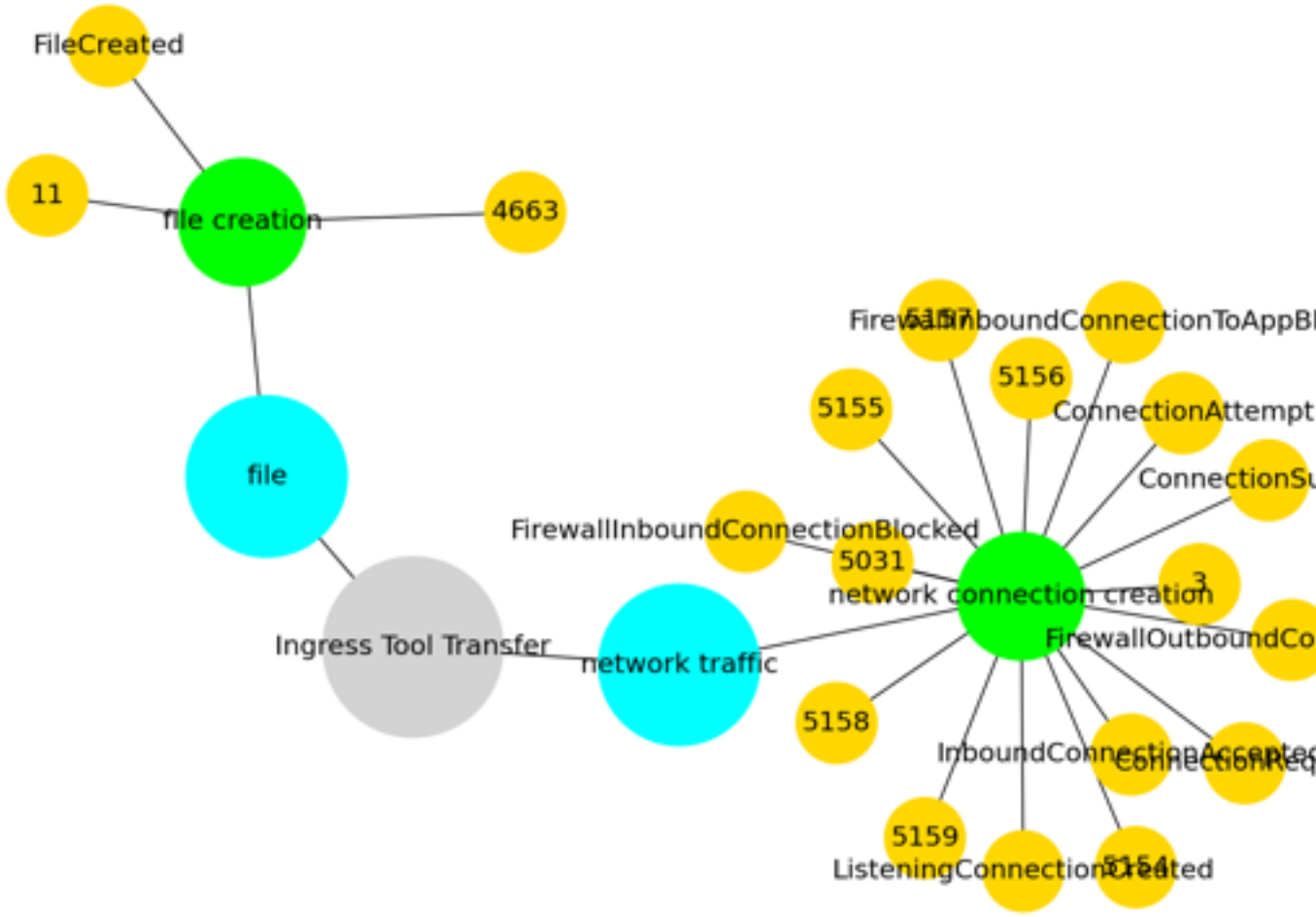
Tactic : command-and-control

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)



3.2 T1204.002

Used by group : Tonto Team, Sharpshooter, APT19, APT33, OilRig, menuPass, Threat Group-3390

Tactic : execution

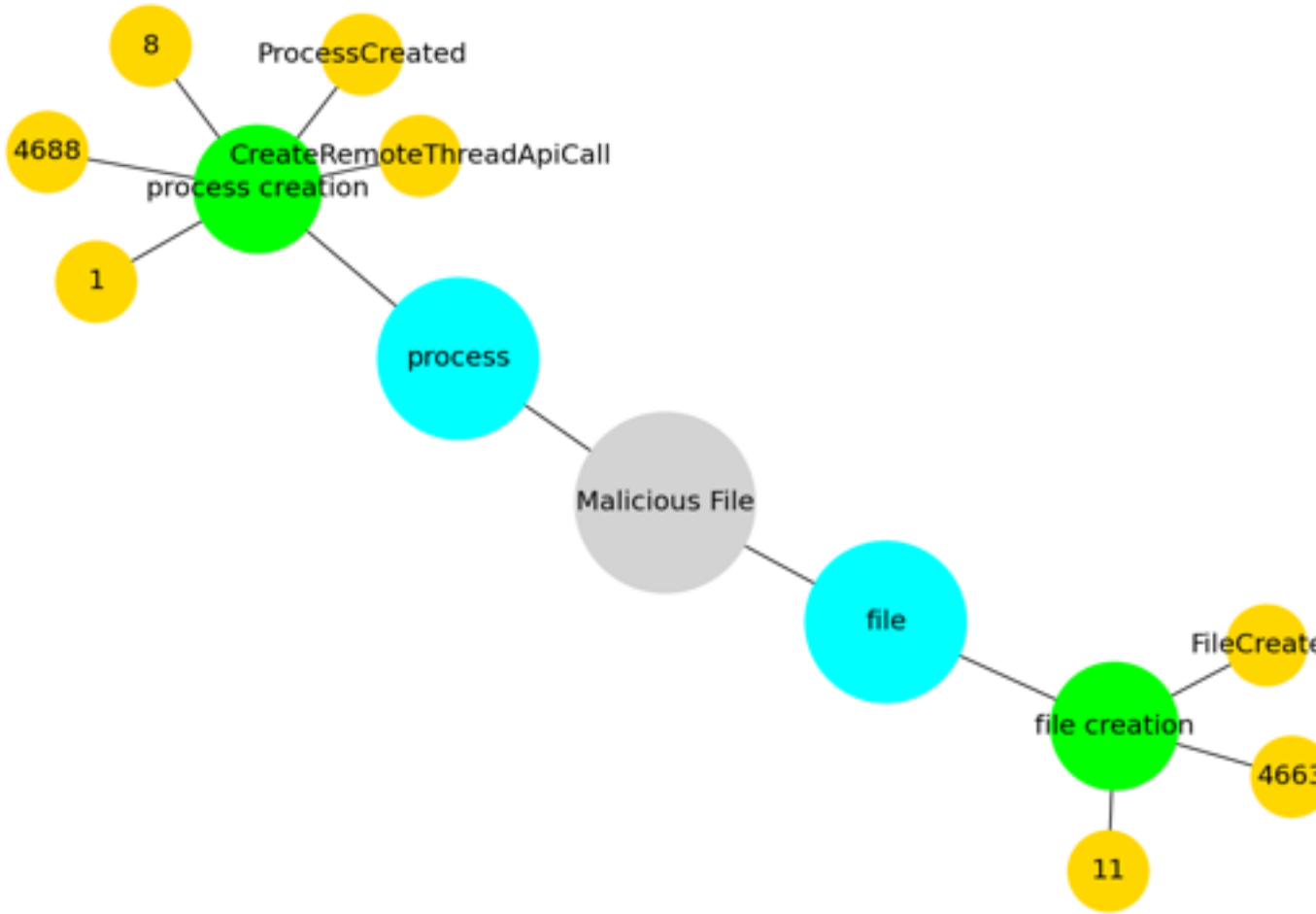
Technique : Malicious File

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).



3.3 T1059.001

Used by group : Tonto Team, Operation Wocao, APT19, APT33, OilRig, menuPass, Threat Group-3390

Tactic : execution

Technique : PowerShell

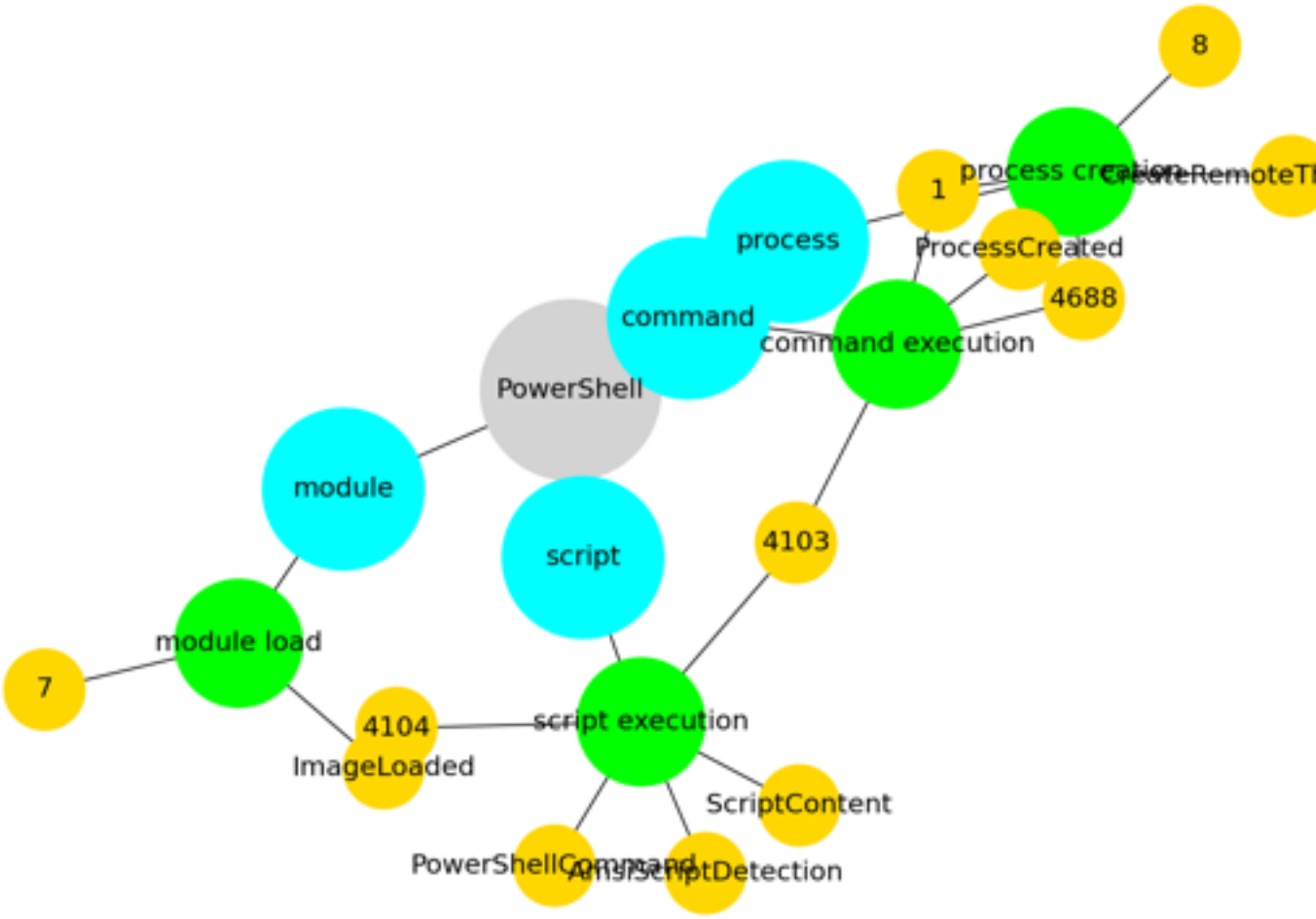
Adversaries may abuse PowerShell commands and scripts for execution. PowerShell is a powerful interactive command-line interface and scripting environment included in the Windows operating system.(Citation: TechNet PowerShell) Adversaries can use PowerShell

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)



3.4 T1566.001

Used by group : Tonto Team, Sharpshooter, APT19, APT33, OilRig, menuPass, Threat Group-3390

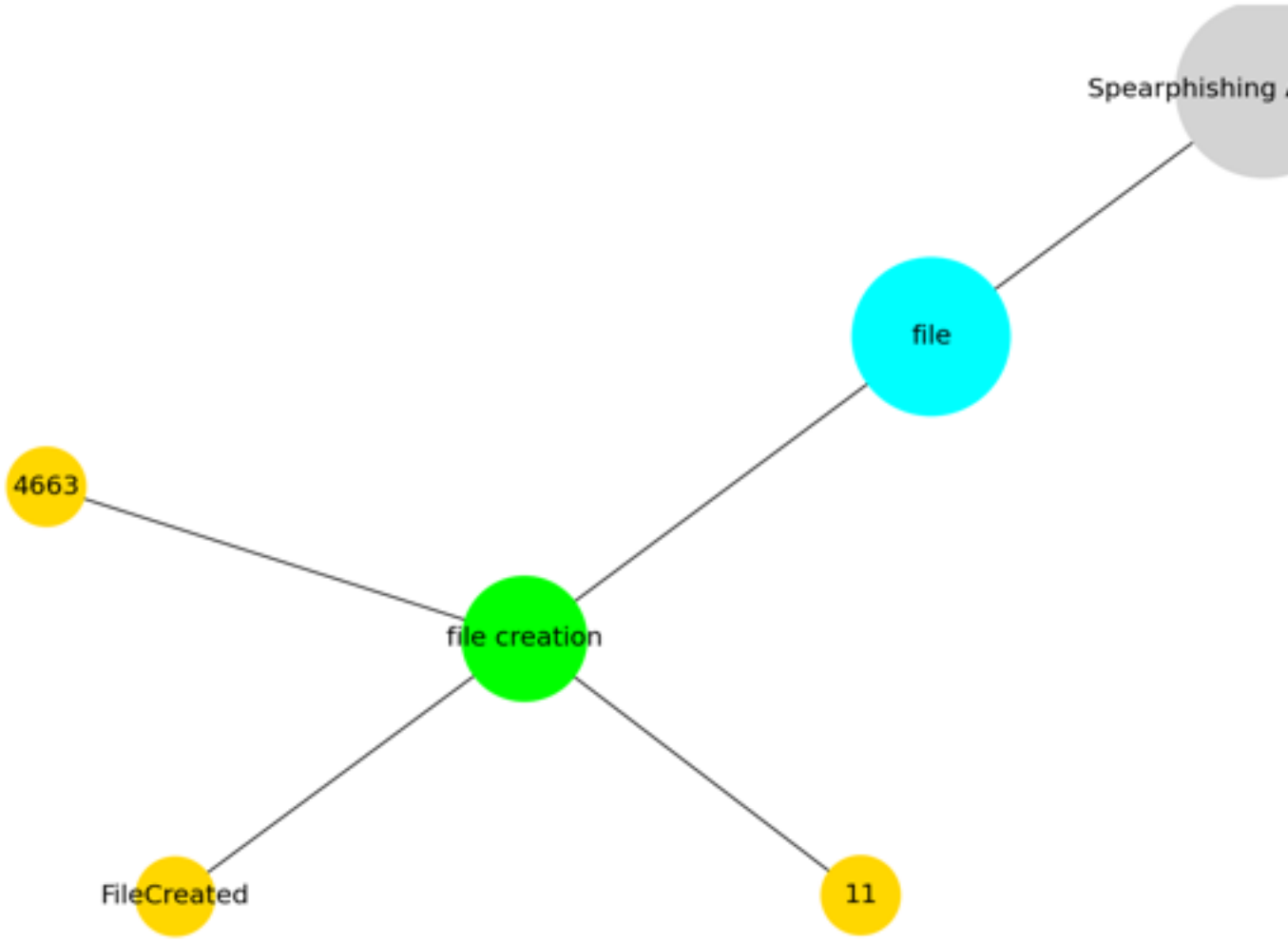
Tactic : initial-access

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.



3.5 T1027

Used by group : Operation Wocao, APT19, APT33, OilRig, menuPass, Threat Group-3390

Tactic : defense-evasion

Technique : Obfuscated Files or Information

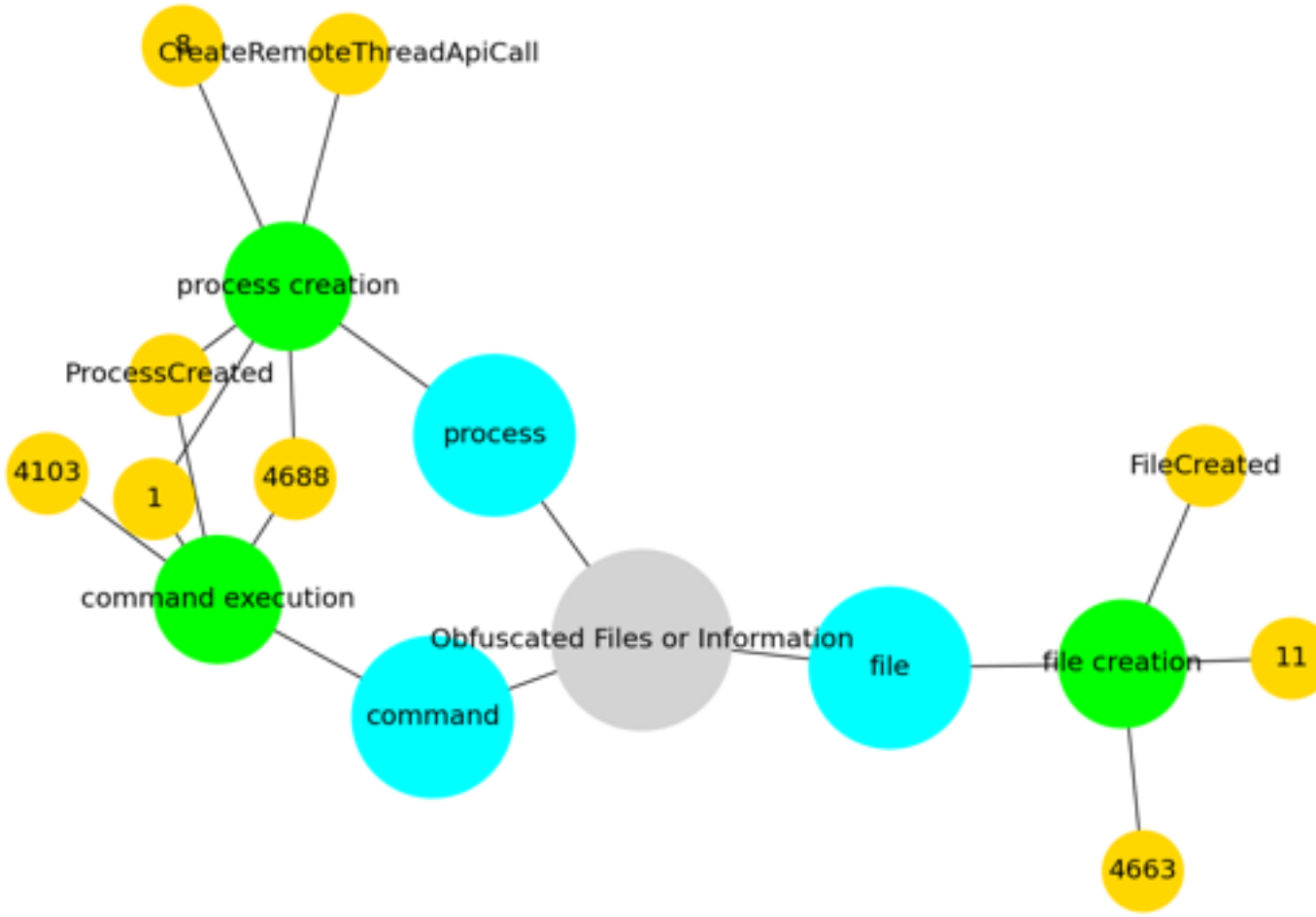
Adversaries may attempt to make an executable or file difficult to discover or analyze by encrypting, encoding, or otherwise

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)



3.6 T1016

Used by group : Operation Wocao, APT19, OilRig, menuPass, Threat Group-3390

Tactic : discovery

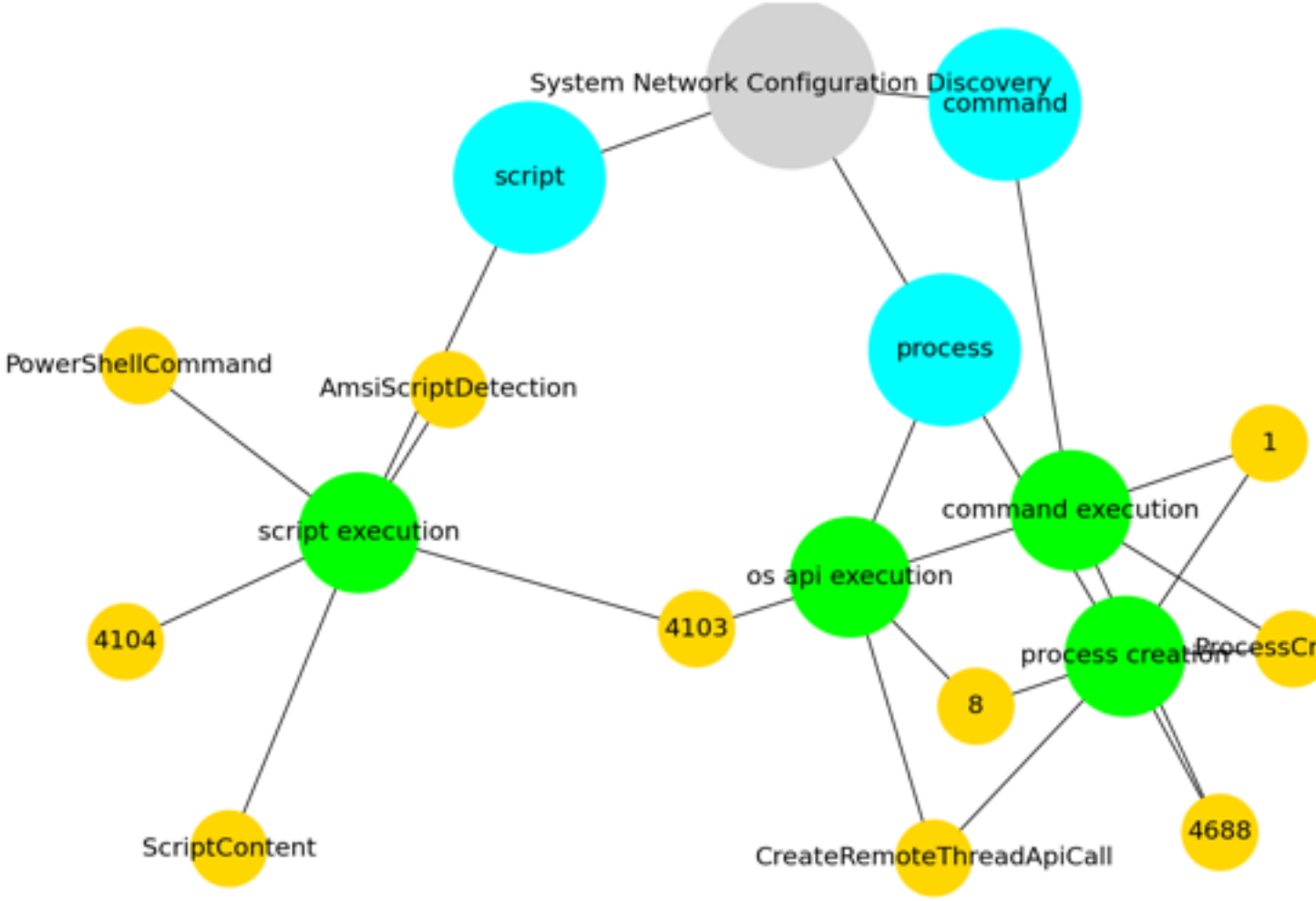
Technique : System Network Configuration Discovery

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.



3.7 T1056.001

Used by group : Tonto Team, Operation Wocao, OilRig, menuPass, Threat Group-3390

Tactic : collection, credential-access

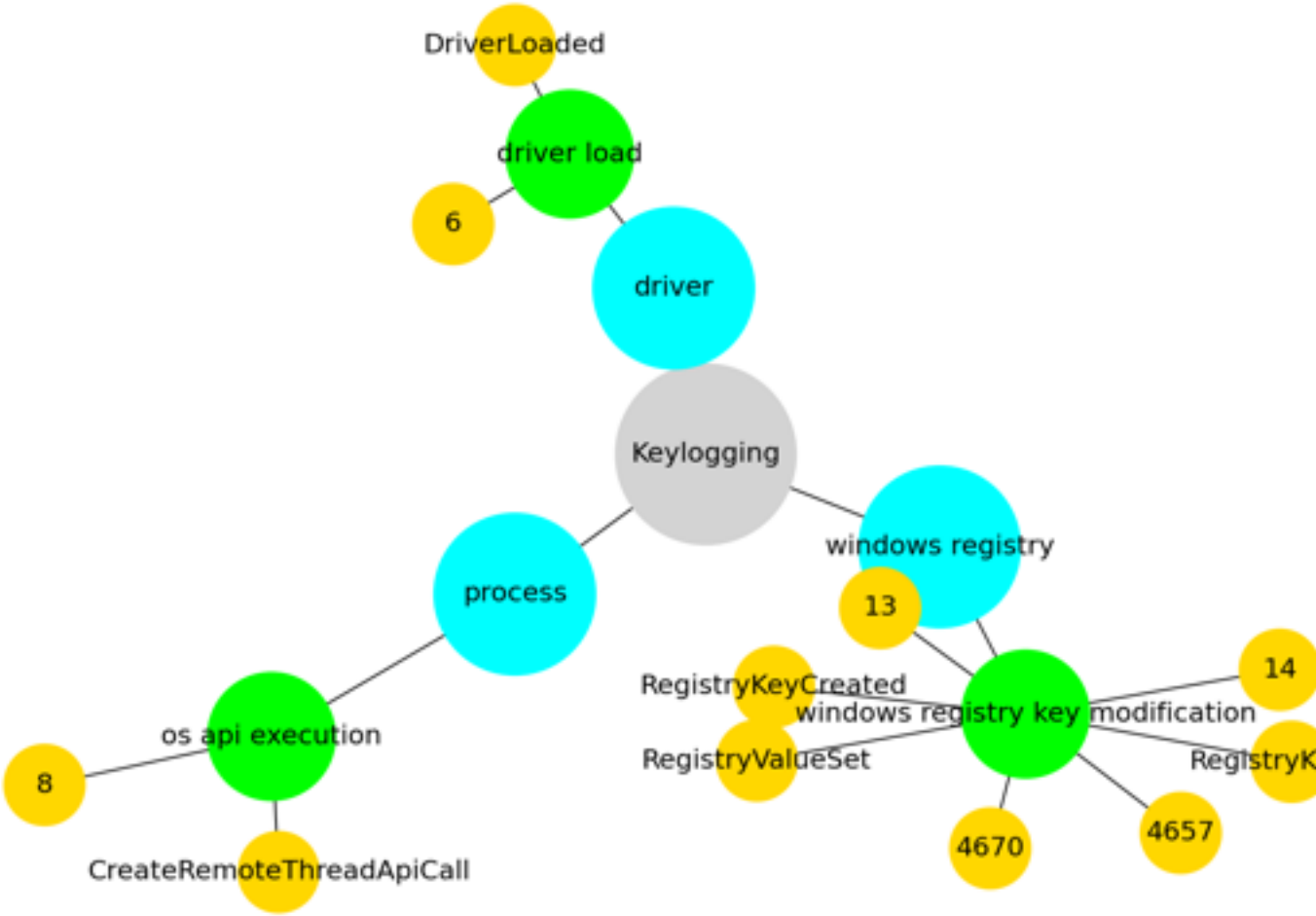
Technique : Keylogging

Adversaries may log user keystrokes to intercept credentials as the user types them. Keylogging is likely to be used to acquire credentials for new access opportunities when [OS Credential Dumping](https://attack.mitre.org/techniques/T1003) efforts are not

effective, and may require an adversary to intercept keystrokes on a system for a substantial period of time before credentials can be successfully captured.

Keylogging is the most prevalent type of input capture, with many different ways of intercepting keystrokes.(Citation: Adventures of a Keystroke) Some methods include:

* Hooking API callbacks used for processing keystrokes. Unlike [Credential API Hooking](https://attack.mitre.org/techniques/T1056/004), this focuses solely on API functions intended for processing keystroke data.
* Reading raw keystroke data from the hardware buffer.
* Windows Registry modifications.
* Custom drivers.
* [Modify System Image](https://attack.mitre.org/techniques/T1601) may provide adversaries with hooks into the operating system of network devices to read raw keystrokes for login sessions.(Citation: Cisco Blog Legacy Device Attacks)



3.8 T1078

Used by group : Operation Wocao, APT33, OilRig, menuPass, Threat Group-3390

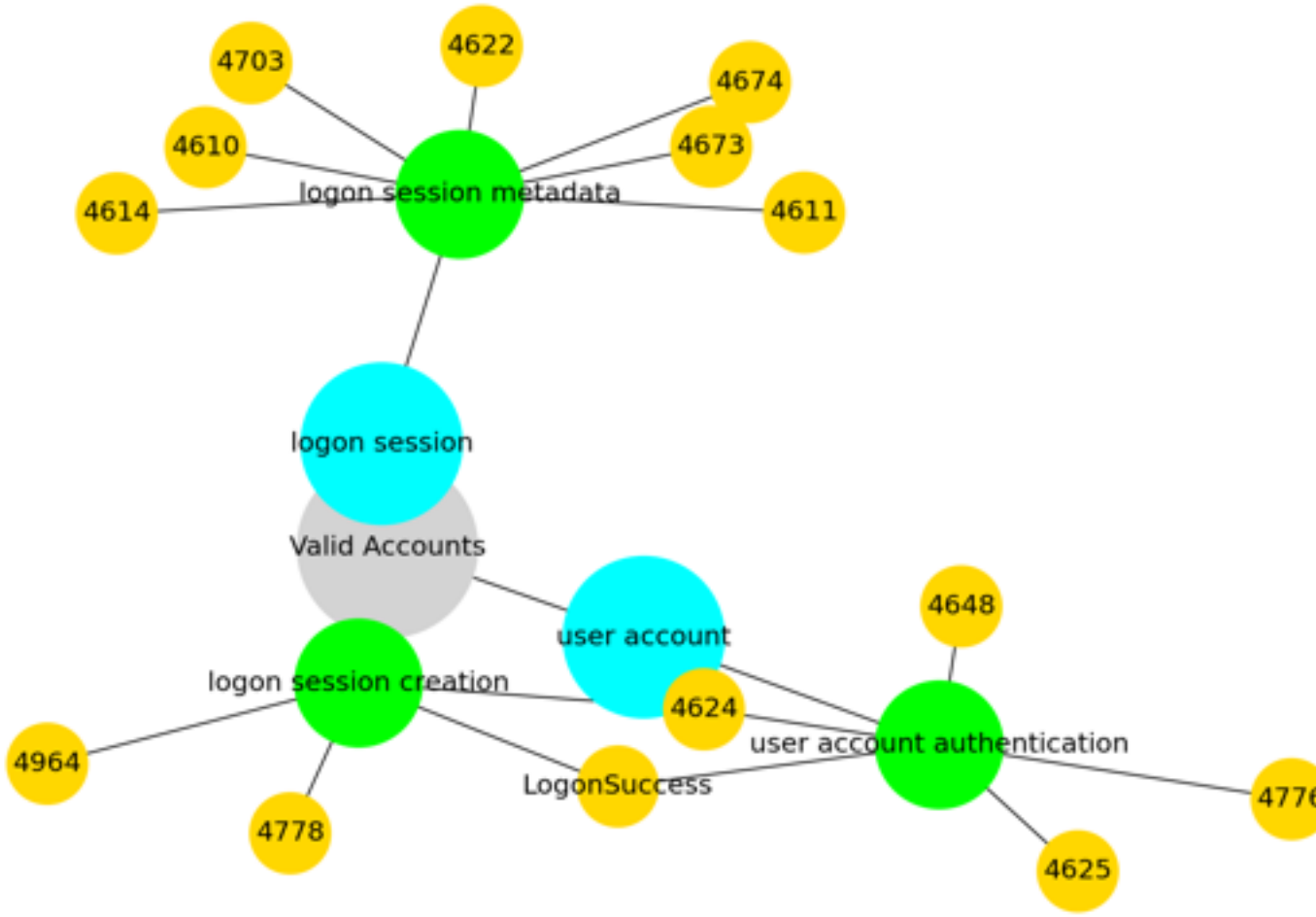
Tactic : defense-evasion, persistence, privilege-escalation, initial-access

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)



3.9 T1003.004

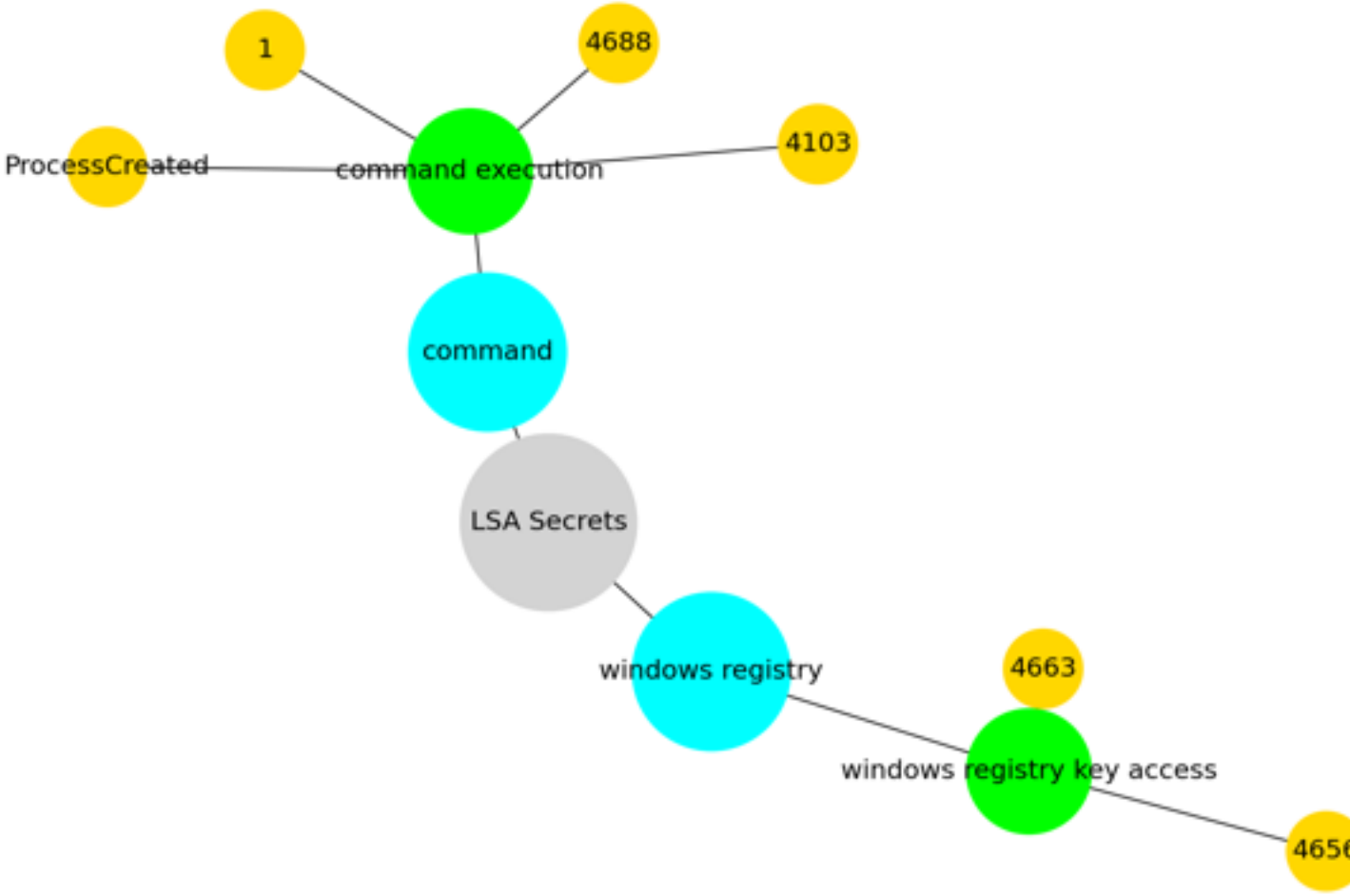
Used by group : APT33, OilRig, menuPass, Threat Group-3390

Tactic : credential-access

Technique : LSA Secrets

Adversaries with SYSTEM access to a host may attempt to access Local Security Authority (LSA) secrets, which can contain a variety of different credential materials, such as credentials for service accounts.(Citation: Passcape LSA Secrets)(Citation: Microsoft AD Admin Tier Model)(Citation: Tilbury Windows Credentials) LSA secrets are stored in the registry at <code>HKEY_LOCAL_MACHINE\SECURITY\Policy\Secrets</code>. LSA secrets can also be dumped from memory.(Citation: ired Dumping LSA Secrets)

[Reg](https://attack.mitre.org/software/S0075) can be used to extract from the Registry. [Mimikatz](https://attack.mitre.org/software/S0002) can be used to extract secrets from memory.(Citation: ired Dumping LSA Secrets)



3.10 T1071.001

Used by group : APT19, APT33, OilRig, Threat Group-3390

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.

3.11 T1119

Used by group : Operation Wocao, OilRig, menuPass, Threat Group-3390

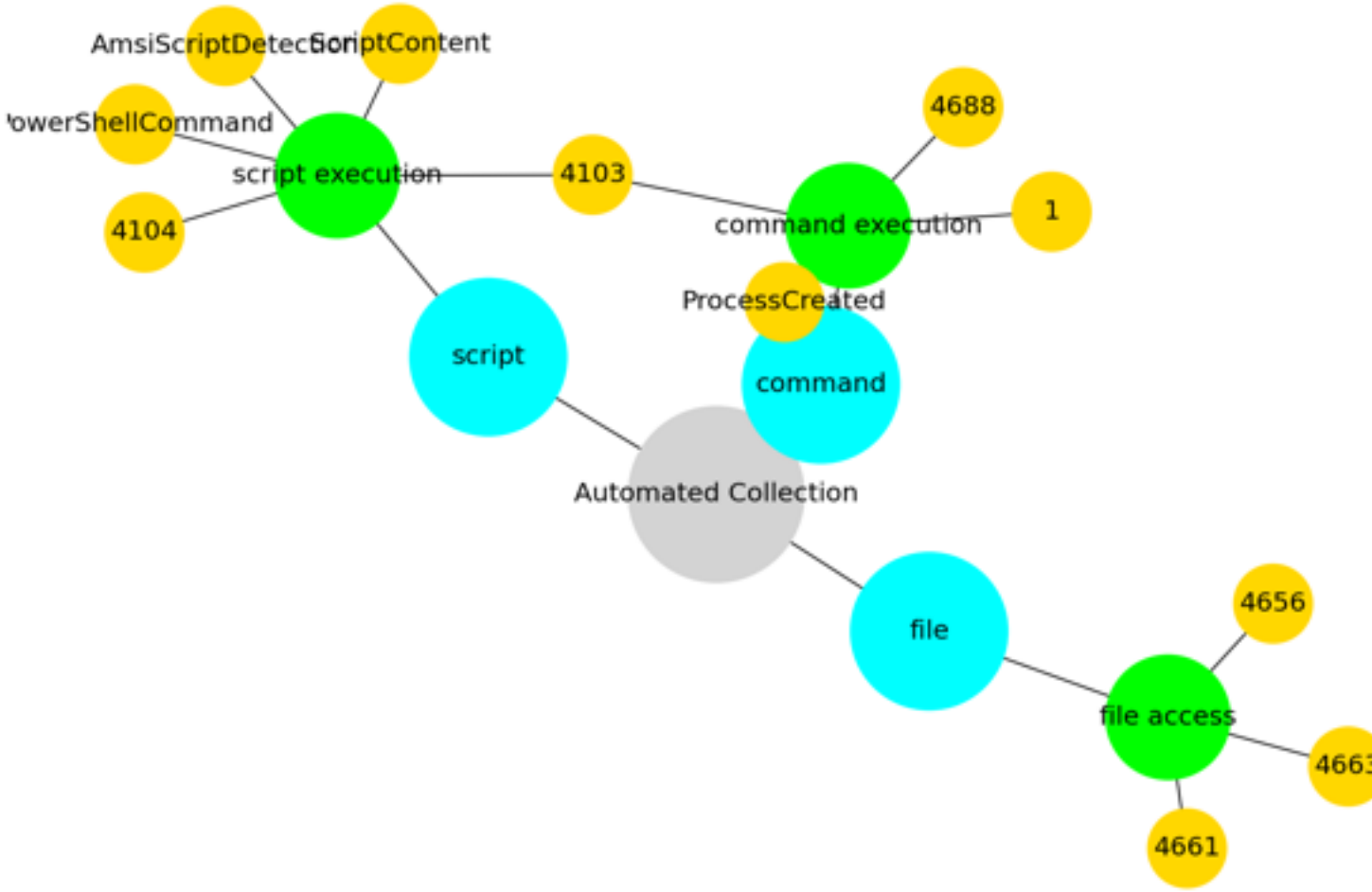
Tactic : collection

Technique : Automated Collection

Once established within a system or network, an adversary may use automated techniques for collecting internal data. Methods for performing this technique could include use of a [Command and Scripting Interpreter](https://attack.mitre.org/techniques/T1059) to search for and copy information fitting set criteria such as file type, location, or name at specific time intervals. In cloud-based environments, adversaries may also use cloud APIs, command line interfaces, or extract, transform, and load (ETL) services to automatically collect data. This functionality could also be built into remote access tools.

This technique may incorporate use of other techniques such as [File and Directory Discovery](https://attack.mitre.org/techniques/T1083) and [Lateral Tool Transfer](https://attack.mitre.org/techniques/T1570) to identify

and move files, as well as [Cloud Service Dashboard](https://attack.mitre.org/techniques/T1538) and [Cloud Storage Object Discovery](https://attack.mitre.org/techniques/T1619) to identify resources in cloud environments.



3.12 T1053.005

Used by group : Operation Wocao, APT33, OilRig, menuPass

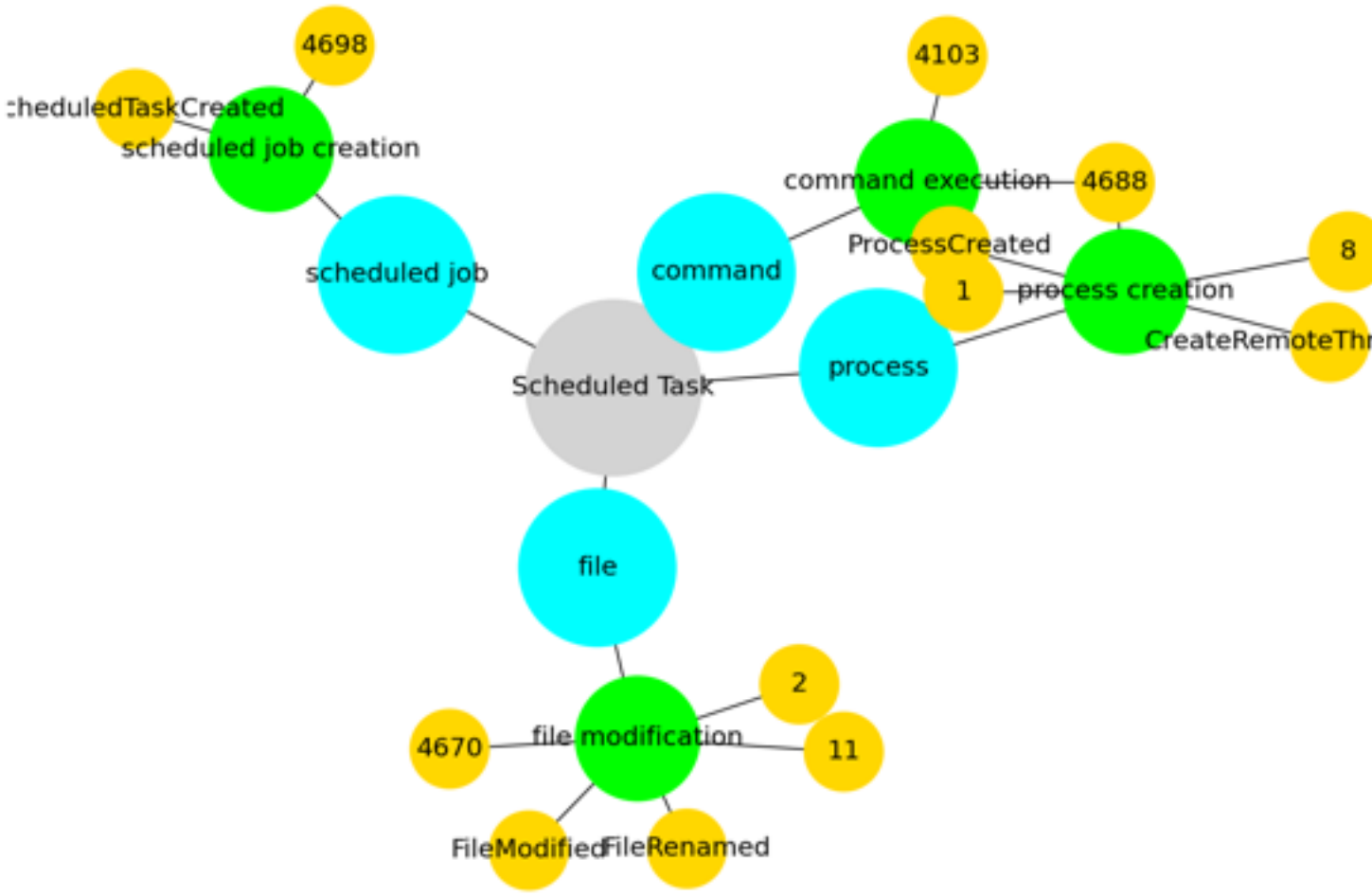
Tactic : execution, persistence, privilege-escalation

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)



3.13 T1059.005

Used by group : Operation Wocao, Sharpshooter, APT33, OilRig

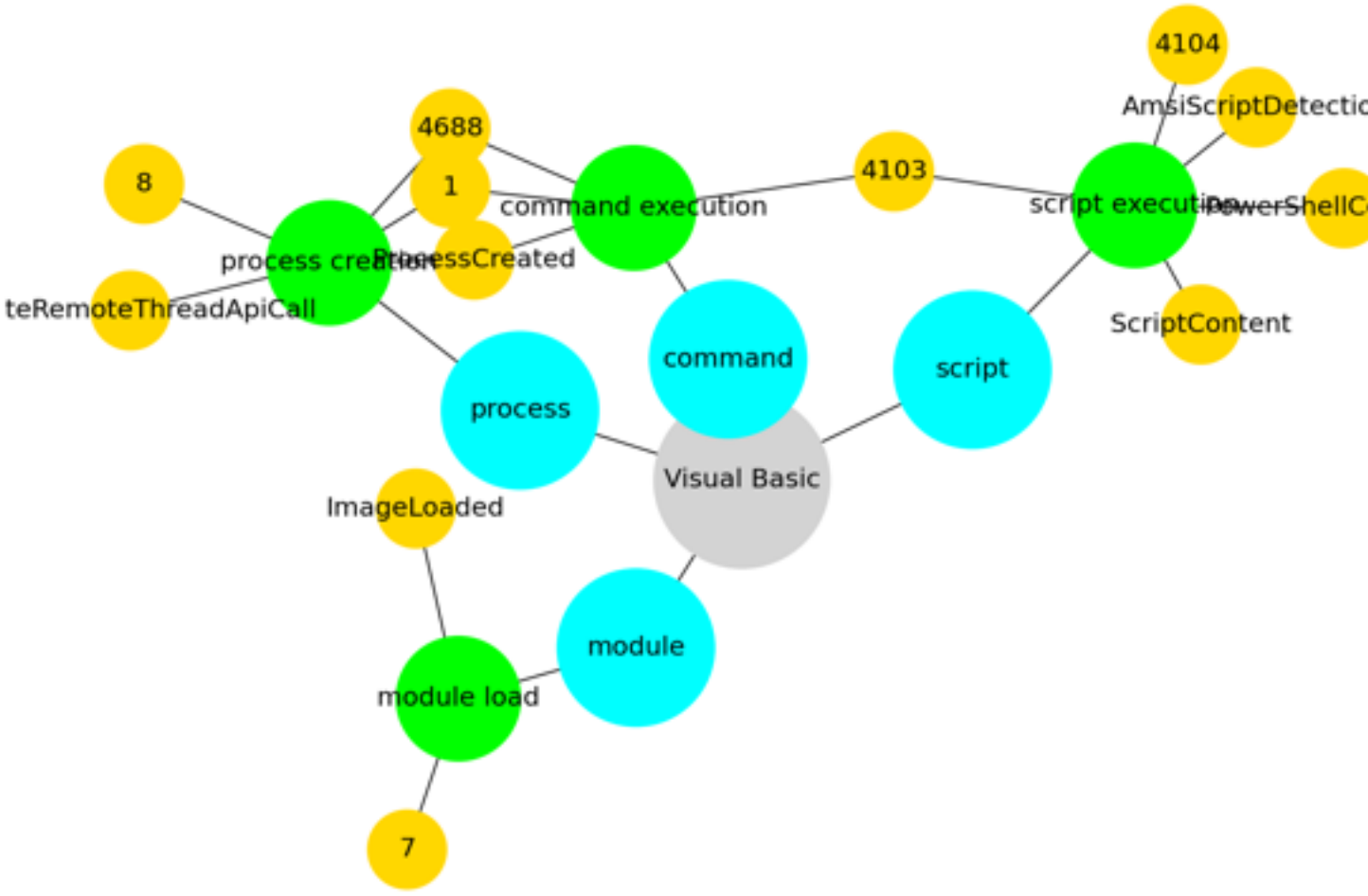
Tactic : execution

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)



3.14 T1059.003

Used by group : Operation Wocao, OilRig, menuPass, Threat Group-3390

Tactic : execution

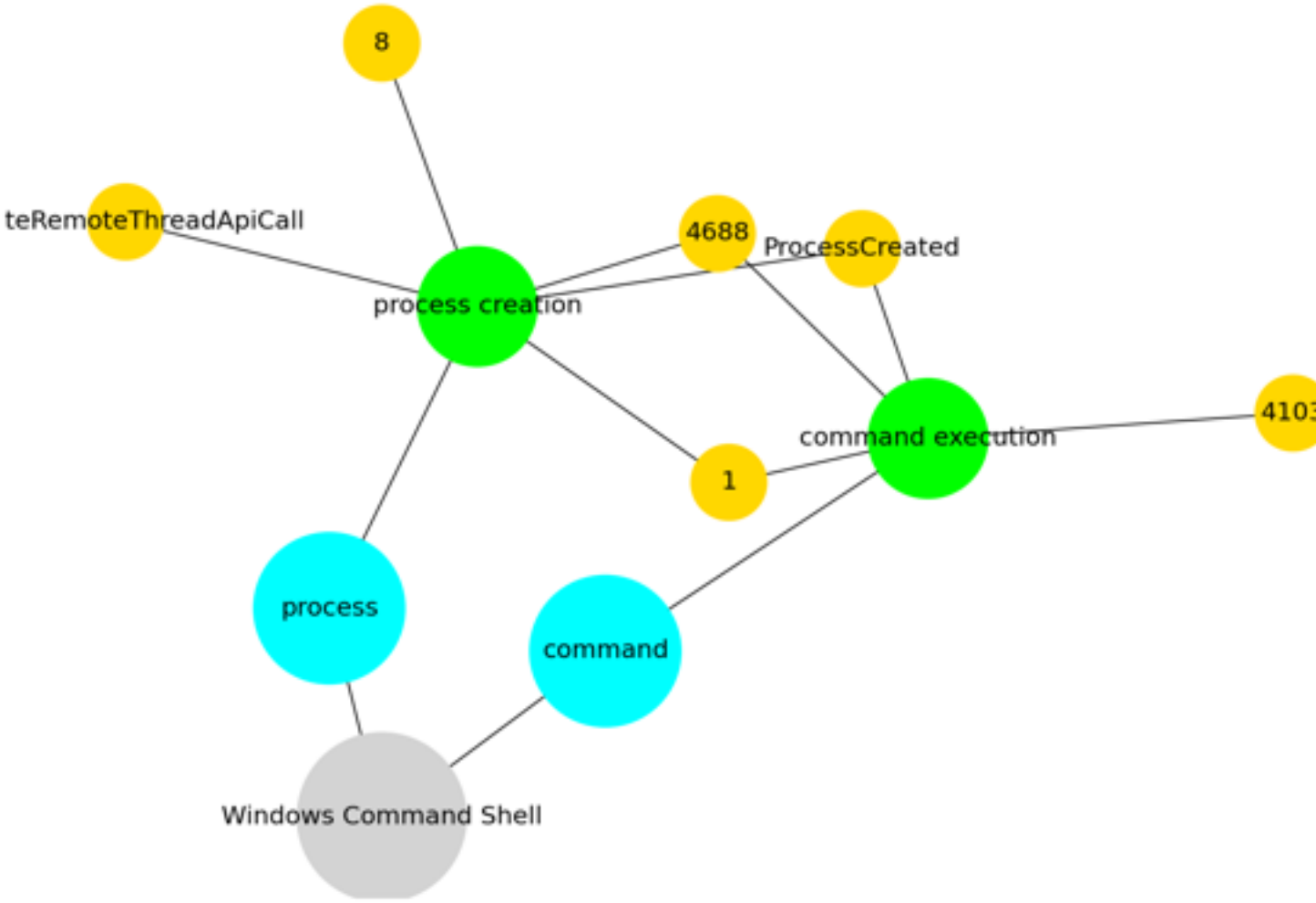
Technique : Windows Command Shell

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.



3.15 T1140

Used by group : APT19, OilRig, menuPass, Threat Group-3390

Tactic : defense-evasion

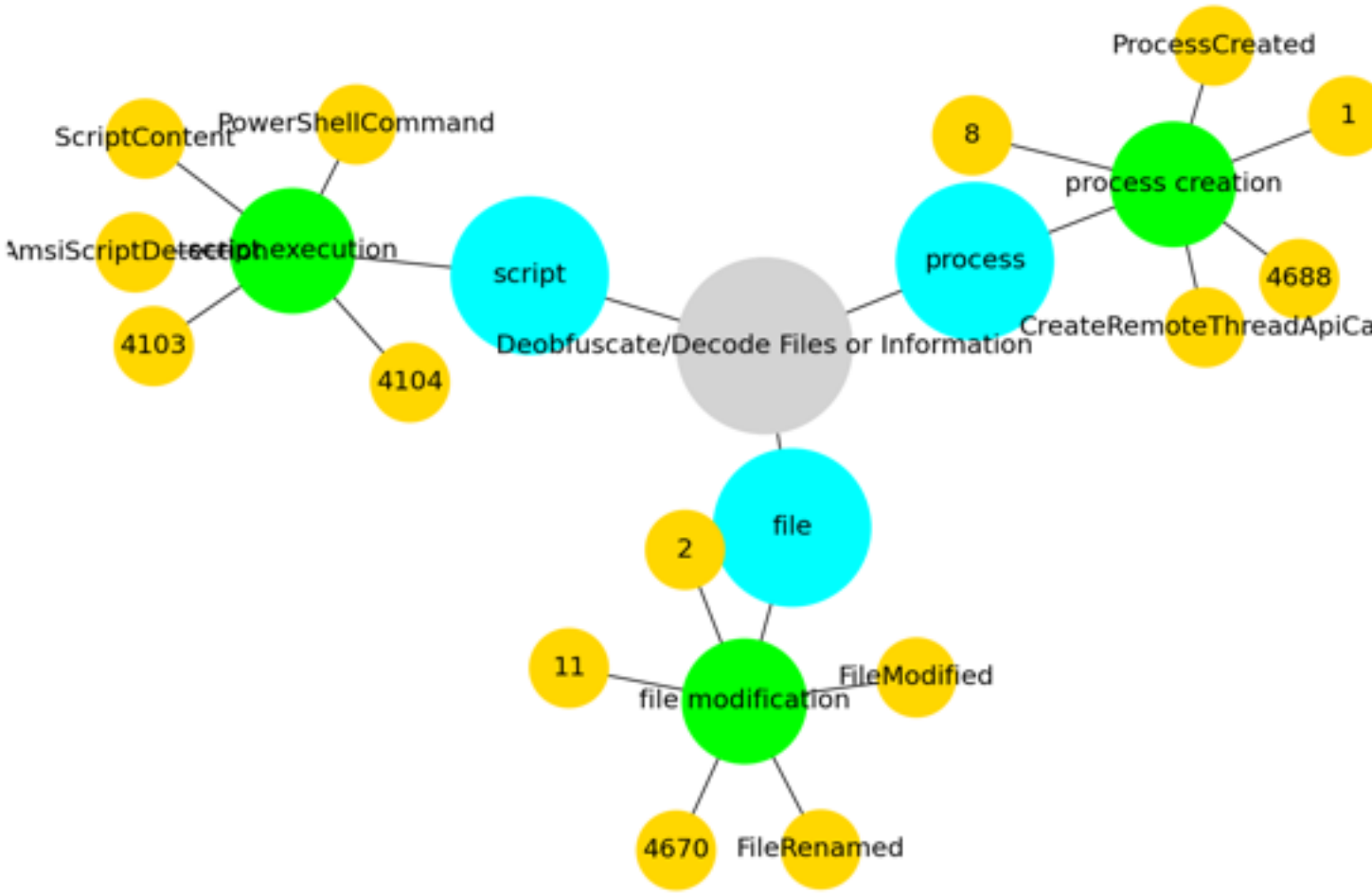
Technique : Deobfuscate/Decode Files or Information

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)



3.16 T1547.001

Used by group : Sharpshooter, APT19, APT33, Threat Group-3390

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>

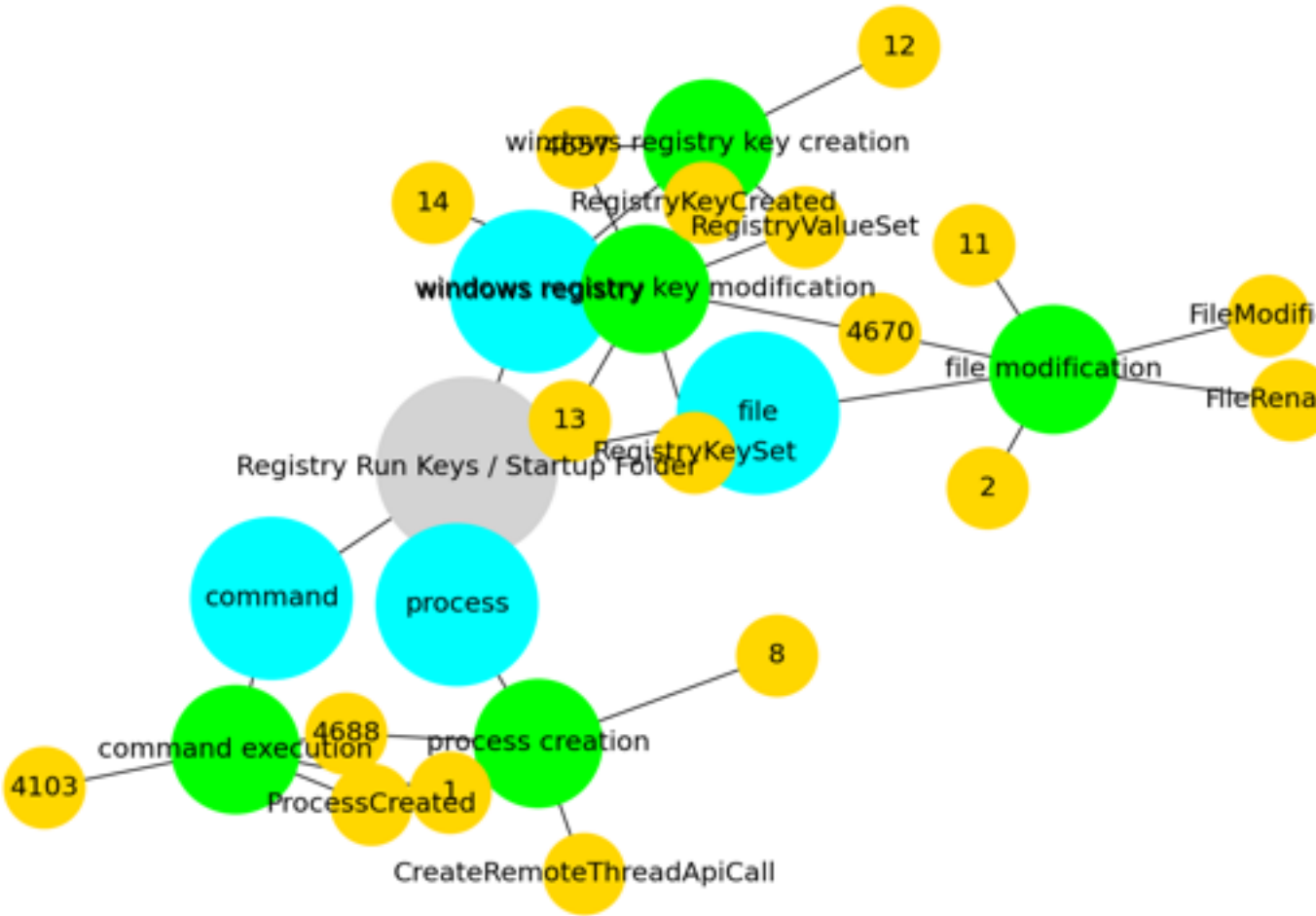
The Winlogon key controls actions that occur when a user logs on to a computer running Windows 7. Most of these actions are under the control of the operating system, but you can also add custom actions here. The <code>HKEY_LOCAL_MACHINE\Software\Microsoft\Windows NT\CurrentVersion\Winlogon\Userinit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



3.17 T1588.002

Used by group : APT19, APT33, menuPass, Threat Group-3390

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).

3.18 T1070.004

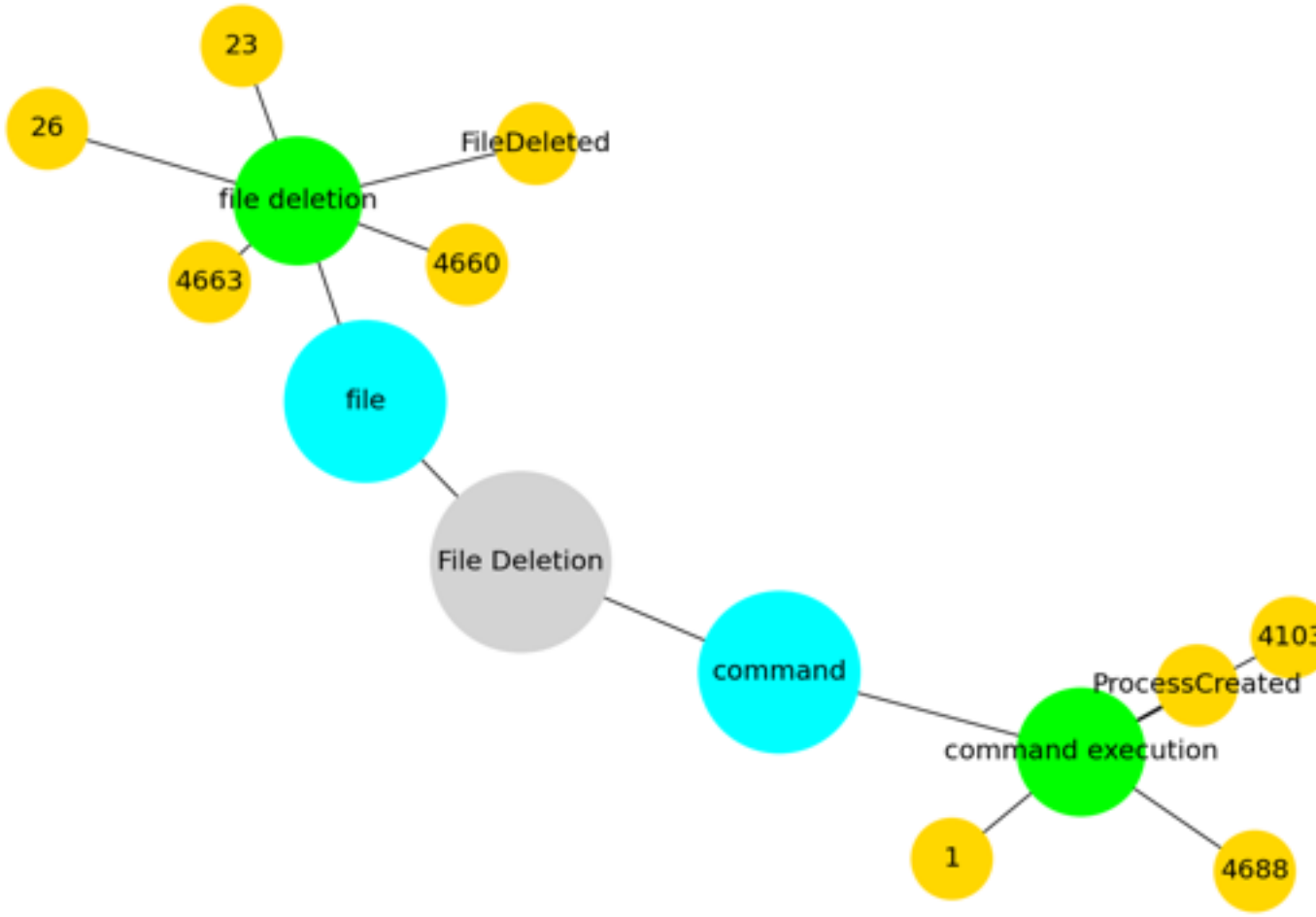
Used by group : Operation Wocao, OilRig, menuPass, Threat Group-3390

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.



3.19 T1003.001

Used by group : Operation Wocao, APT33, OilRig, Threat Group-3390

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>

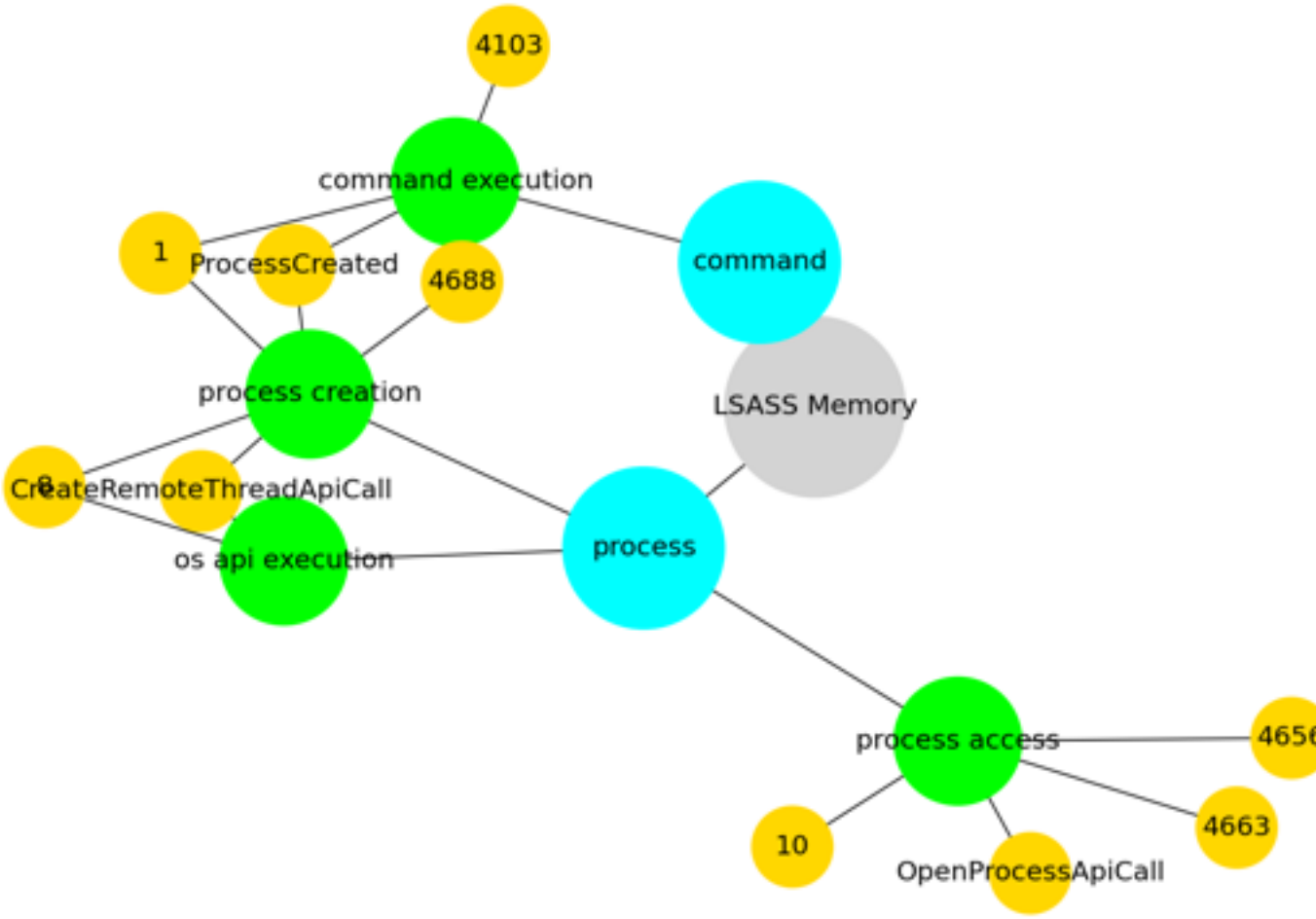
Built-in Windows tools such as comsvcs.dll can also be used:

* <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)



3.20 T1049

Used by group : Operation Wocao, OilRig, menuPass, Threat Group-3390

Tactic : discovery

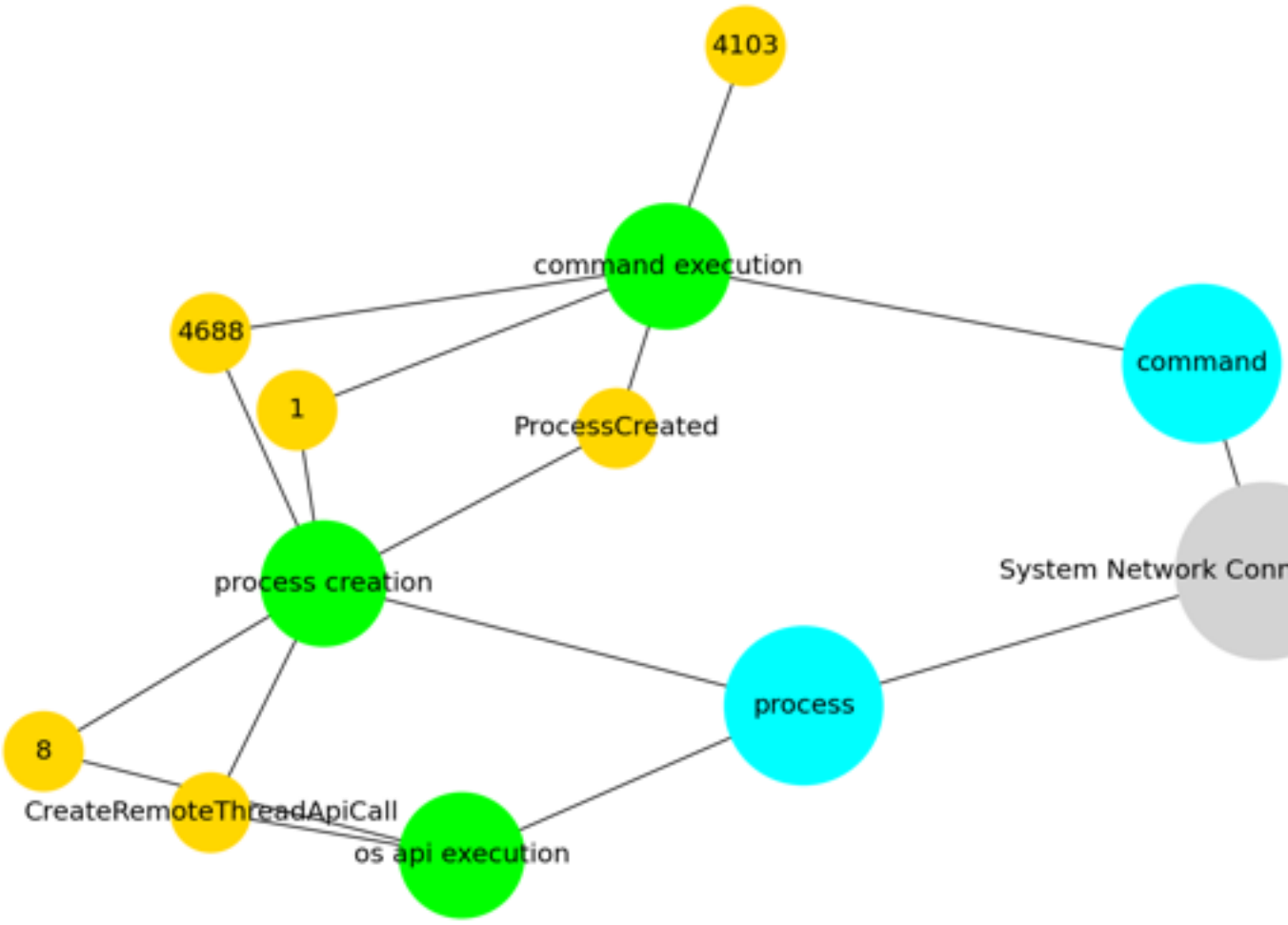
Technique : System Network Connections Discovery

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)



3.21 T1047

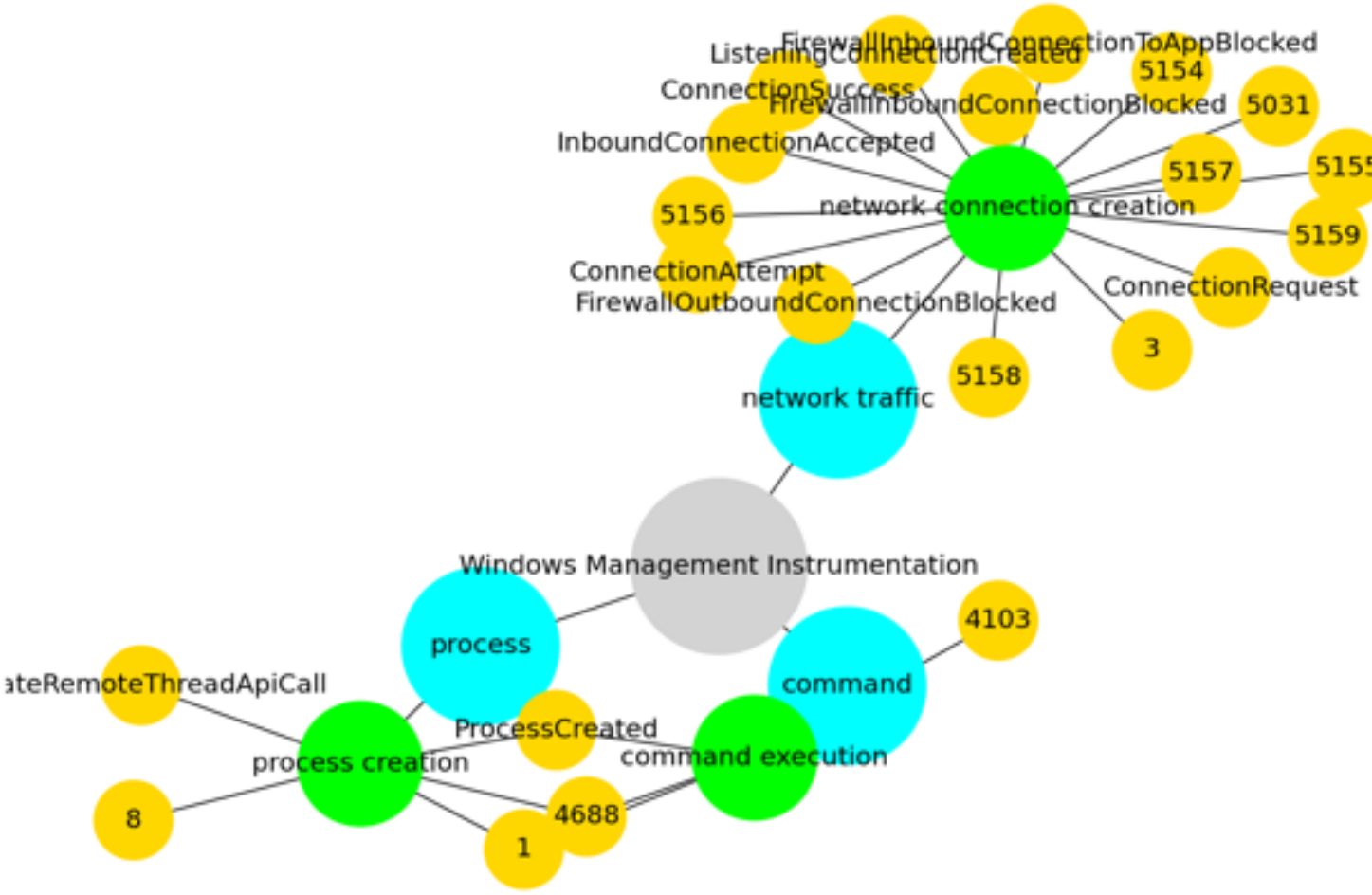
Used by group : Operation Wocao, OilRig, menuPass, Threat Group-3390

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)

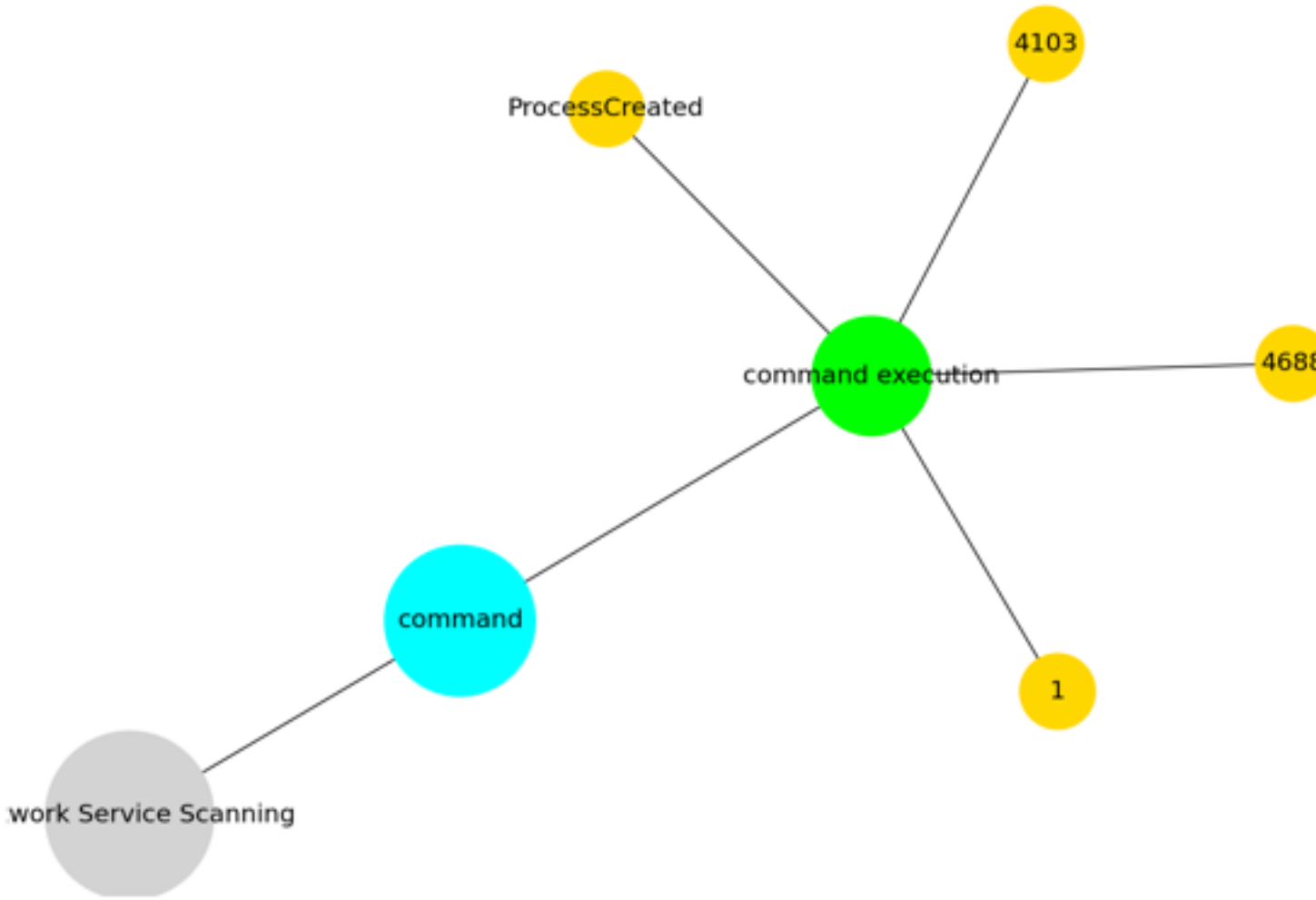
An adversary can use WMI to interact with local and remote systems and use it as a means to execute various behaviors, such as gathering information for Discovery as well as remote Execution of files as part of Lateral Movement. (Citation: FireEye WMI SANS 2015) (Citation: FireEye WMI 2015)



3.22 T1046

3.22 T1046

< To be corrected or added in future releases >



3.23 T1505.003

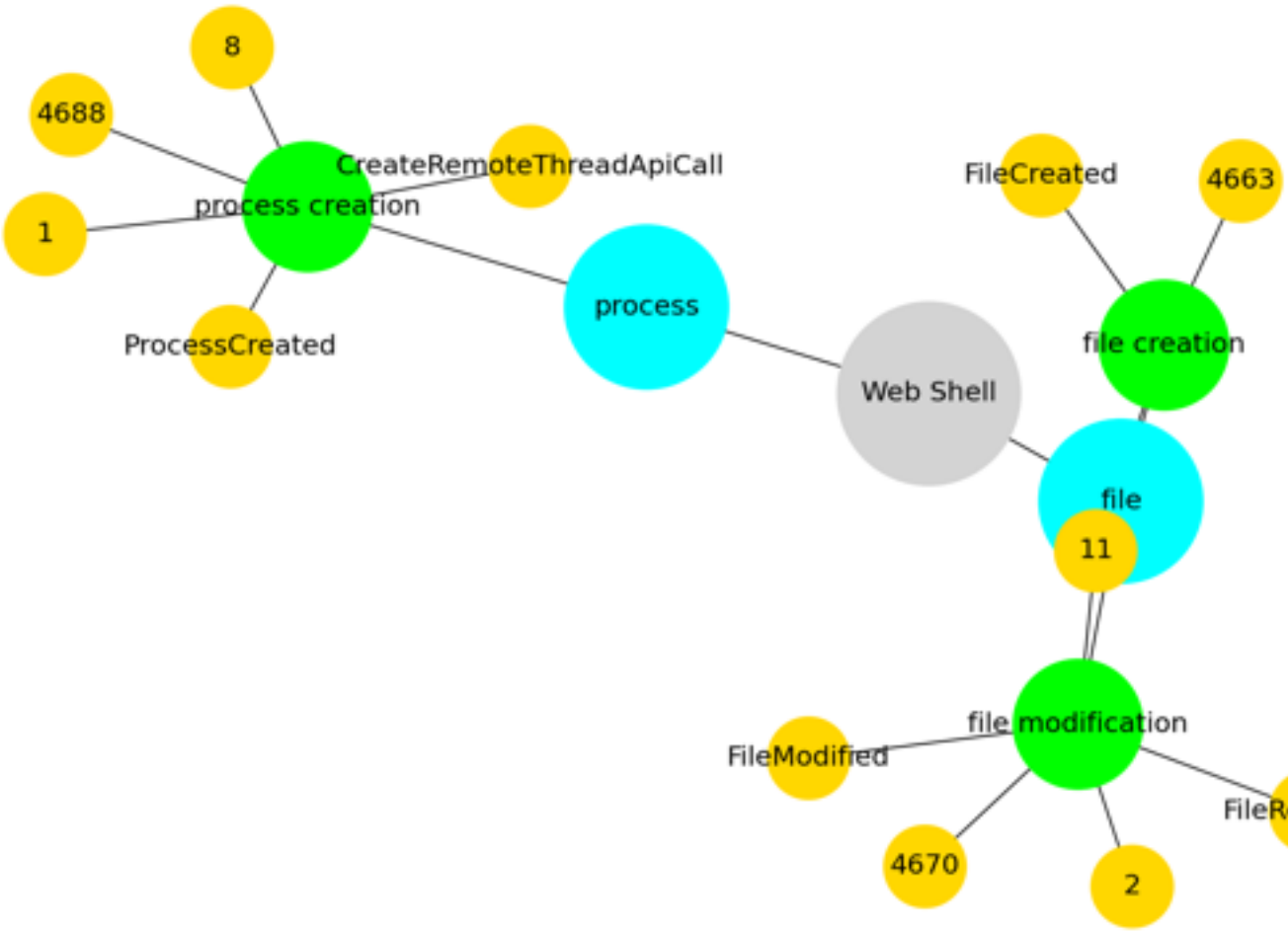
Used by group : Tonto Team, Operation Wocao, OilRig, Threat Group-3390

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)



3.24 T1033

Used by group : Operation Wocao, APT19, OilRig, Threat Group-3390

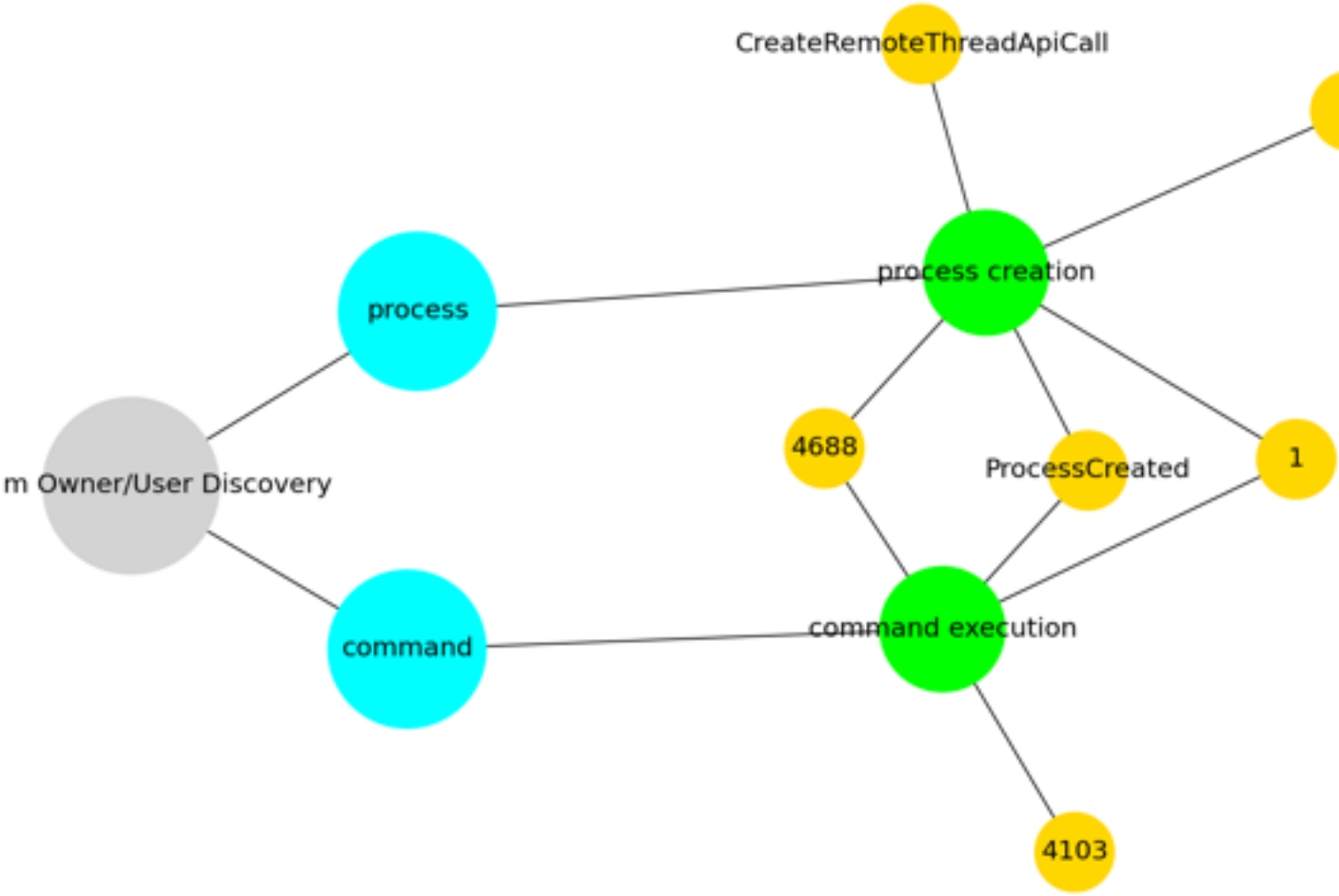
Tactic : discovery

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.



3.25 T1005

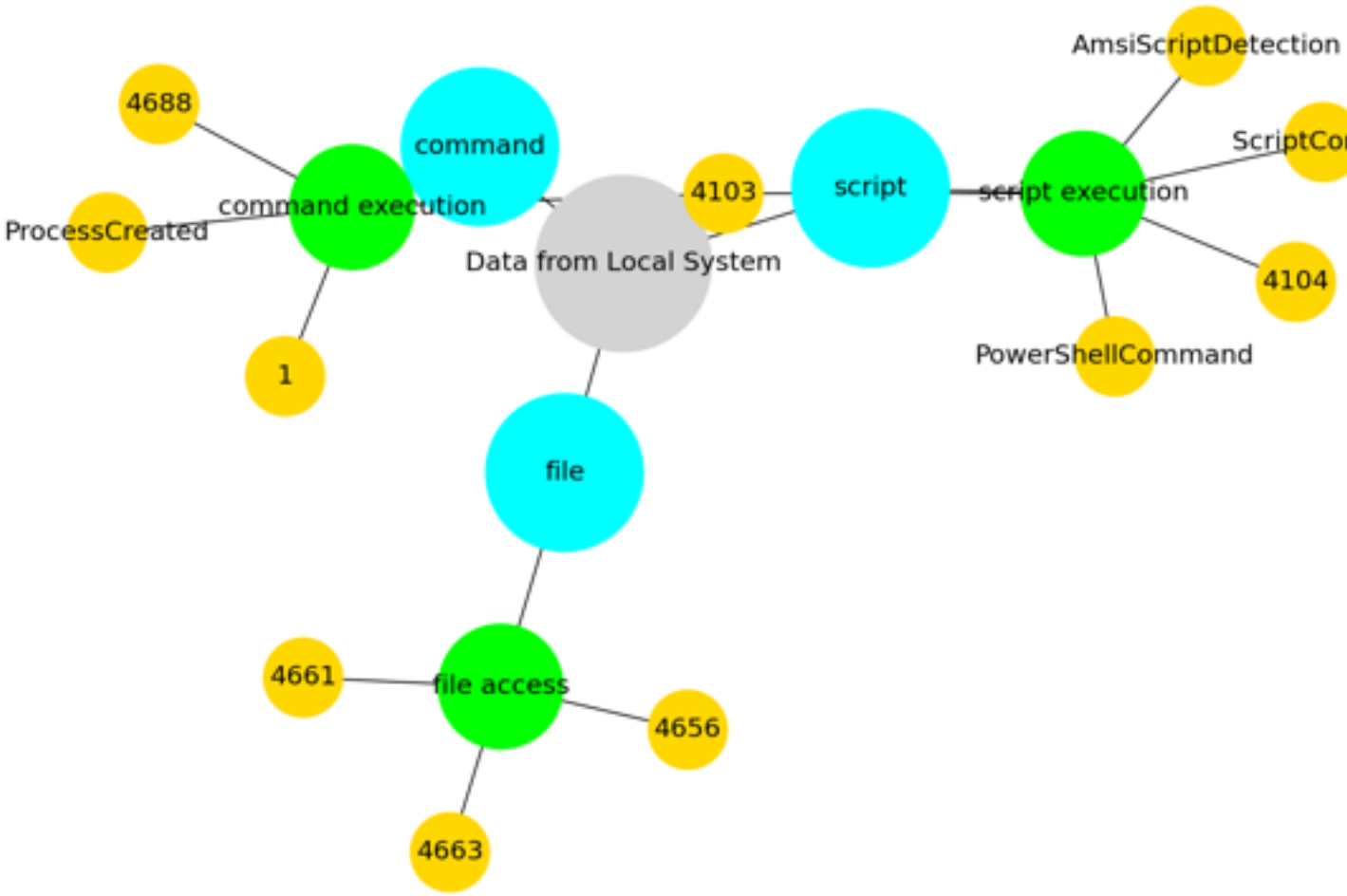
Used by group : Operation Wocao, menuPass, Threat Group-3390

Tactic : collection

Technique : Data from Local System

Adversaries may search local system sources, such as file systems and configuration files or local databases, to find files of interest and sensitive data prior to Exfiltration.

Adversaries may do this using a [Command and Scripting Interpreter](https://attack.mitre.org/techniques/T1059), such as [cmd](https://attack.mitre.org/software/S0106) as well as a [Network Device CLI](https://attack.mitre.org/techniques/T1059/008), which have functionality to interact with the file system to gather information. Adversaries may also use [Automated Collection](https://attack.mitre.org/techniques/T1119) on the local system.



3.26 T1106

Used by group : Operation Wocao, Sharpshooter, menuPass

Tactic : execution

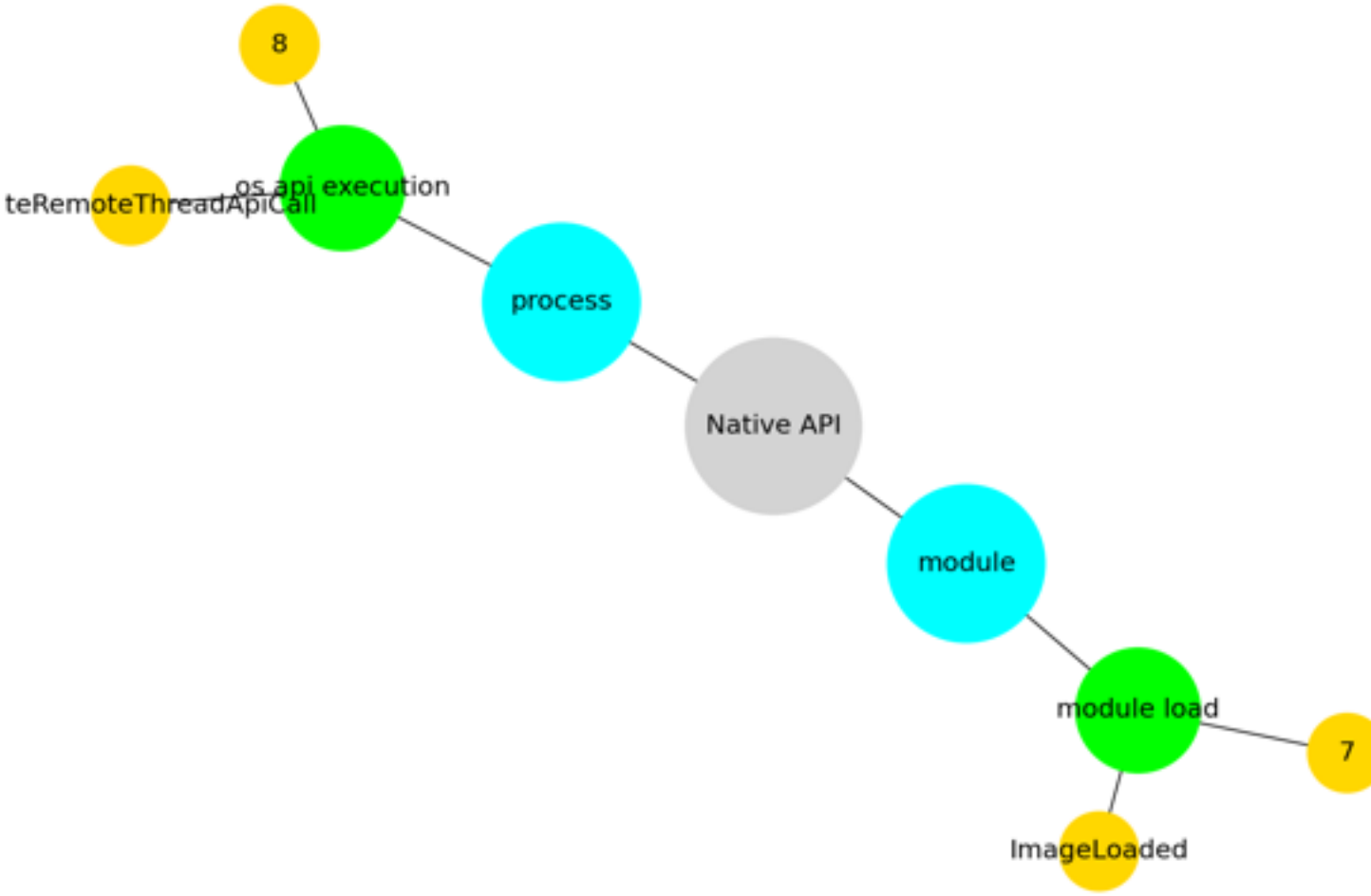
Technique : Native API

Adversaries may interact with the native OS application programming interface (API) to execute behaviors. Native APIs provide a controlled means of calling low-level OS services within the kernel, such as those involving hardware/devices, memory, and processes.(Citation: NT API Windows)(Citation: Linux Kernel API) These native APIs are leveraged by the OS during system boot (when other system components are not yet initialized) as well as carrying out tasks and requests during routine operations.

Native API functions (such as <code>NtCreateProcess</code>) may be directed invoked via system calls / syscalls, but these features are also often exposed to user-mode applications via interfaces and libraries.(Citation: OutFlank System Calls)(Citation: CyberBit System Calls)(Citation: MDSec System Calls) For example, functions such as the Windows API <code>CreateProcess()</code> or GNU <code>fork()</code> will allow programs and scripts to start other processes.(Citation: Microsoft CreateProcess)(Citation: GNU Fork) This may allow API callers to execute a binary, run a CLI command, load modules, etc. as thousands of similar API functions exist for various system operations.(Citation: Microsoft Win32)(Citation: LIBC)(Citation: GLIBC)

Higher level software frameworks, such as Microsoft .NET and macOS Cocoa, are also available to interact with native APIs. These frameworks typically provide language wrappers/abstractions to API functionalities and are designed for ease-of-use/portability of code.(Citation: Microsoft NET)(Citation: Apple Core Services)(Citation: MACOS Cocoa)(Citation: macOS Foundation)

Adversaries may abuse these OS API functions as a means of executing behaviors. Similar to [Command and Scripting Interpreter](https://attack.mitre.org/techniques/T1059), the native API and its hierarchy of interfaces provide mechanisms to interact with and utilize various components of a victimized system. While invoking API functions, adversaries may also attempt to bypass defensive tools (ex: unhooking monitored functions via [Disable or Modify Tools](https://attack.mitre.org/techniques/T1562/001)).



3.27 T1574.001

Used by group : Tonto Team, menuPass, Threat Group-3390

Tactic : persistence, privilege-escalation, defense-evasion

Technique : DLL Search Order Hijacking

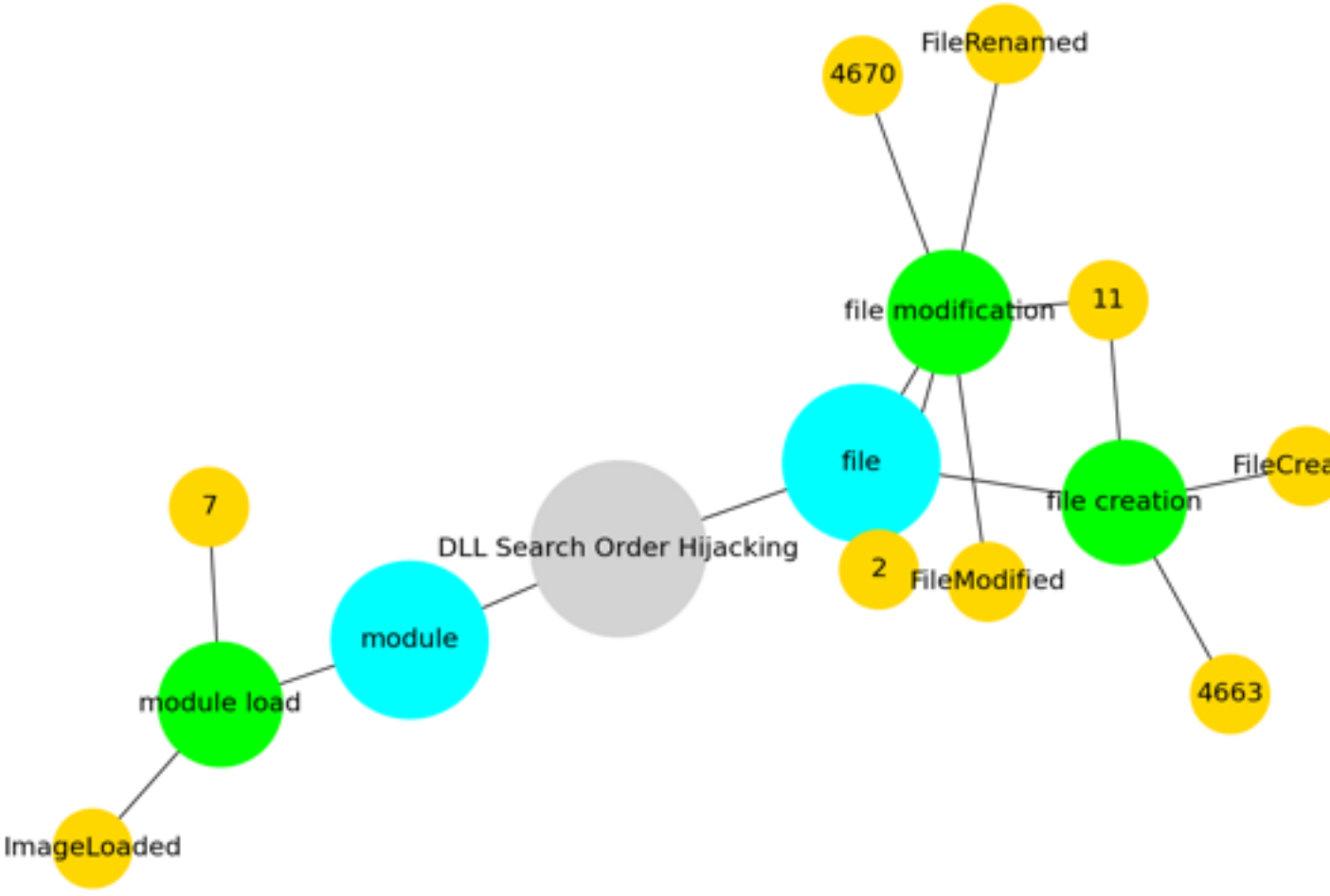
Adversaries may execute their own malicious payloads by hijacking the search order used to load DLLs. Windows systems use a common method to look for required DLLs to load into a program. (Citation: Microsoft Dynamic Link Library Search Order)(Citation: FireEye Hijacking July 2010) Hijacking DLL loads may be for the purpose of establishing persistence as well as elevating privileges and/or evading restrictions on file execution.

There are many ways an adversary can hijack DLL loads. Adversaries may plant trojan dynamic-link library files (DLLs) in a directory that will be searched before the location of a legitimate library that will be requested by a program, causing Windows to load their malicious library when it is called for by the victim program. Adversaries may also perform DLL preloading, also called binary planting

attacks, (Citation: OWASP Binary Planting) by placing a malicious DLL with the same name as an ambiguously specified DLL in a location that Windows searches before the legitimate DLL. Often this location is the current working directory of the program.(Citation: FireEye fxsst June 2011) Remote DLL preloading attacks occur when a program sets its current directory to a remote location such as a Web share before loading a DLL. (Citation: Microsoft Security Advisory 2269637)

Adversaries may also directly modify the search order via DLL redirection, which after being enabled (in the Registry and creation of a redirection file) may cause a program to load a different DLL.(Citation: Microsoft Dynamic-Link Library Redirection)(Citation: Microsoft Manifests)(Citation: FireEye DLL Search Order Hijacking)

If a search order-vulnerable program is configured to run at a higher privilege level, then the adversary-controlled DLL that is loaded will also be executed at the higher level. In this case, the technique could be used for privilege escalation from user to administrator or SYSTEM or from administrator to SYSTEM, depending on the program. Programs that fall victim to path hijacking may appear to behave normally because malicious DLLs may be configured to also load the legitimate DLLs they were meant to replace.



3.28 T1112

Used by group : Operation Wocao, APT19, Threat Group-3390

Tactic : defense-evasion

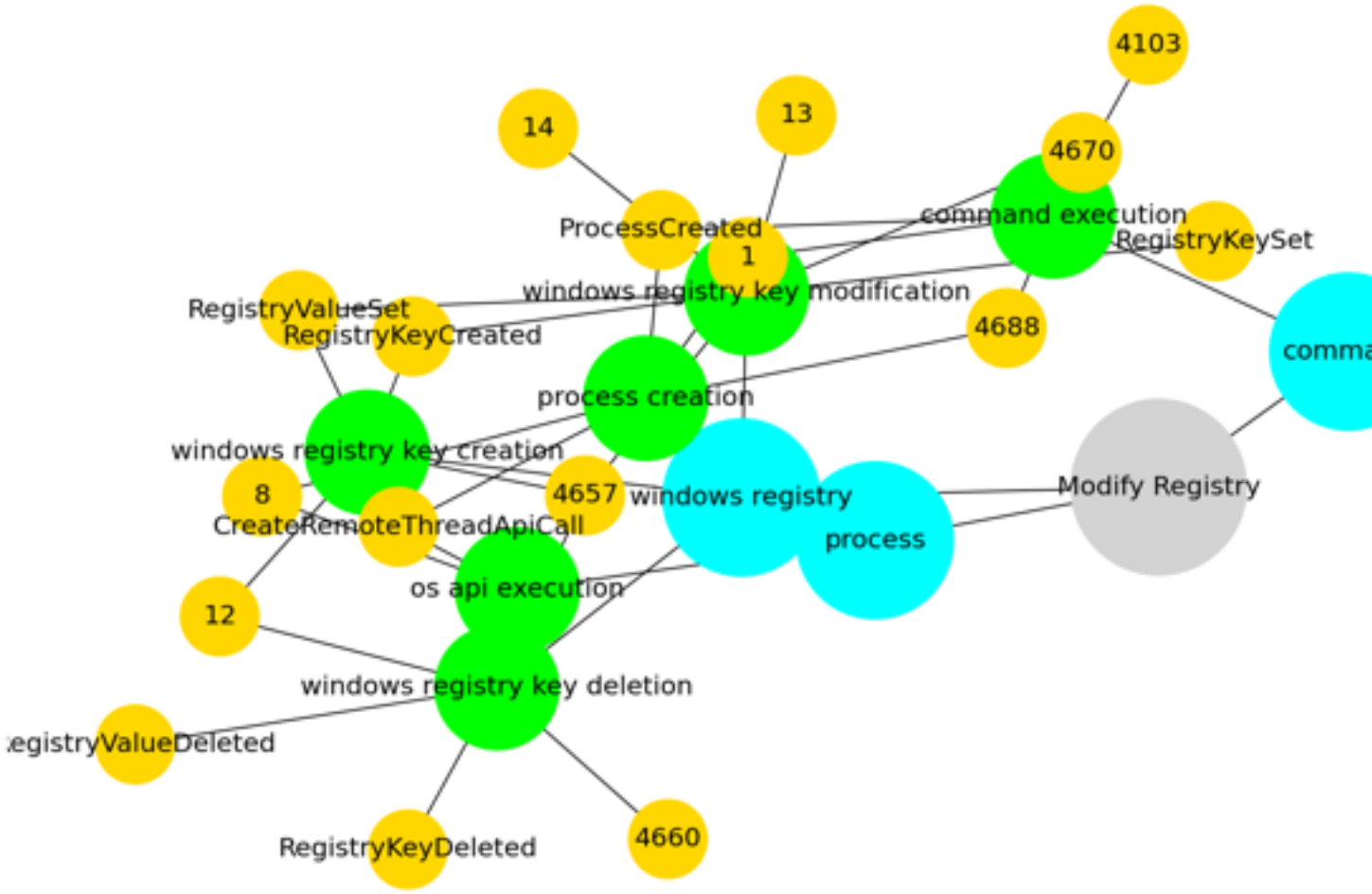
Technique : Modify Registry

Adversaries may interact with the Windows Registry to hide configuration information within Registry keys, remove information as part of cleaning up, or as part of other techniques to aid in persistence and execution.

Access to specific areas of the Registry depends on account permissions, some requiring administrator-level access. The built-in Windows command-line utility [Reg](https://attack.mitre.org/software/S0075) may be used for local or remote Registry modification. (Citation: Microsoft Reg) Other tools may also be used, such as a remote access tool, which may contain functionality to interact with the Registry through the Windows API.

Registry modifications may also include actions to hide keys, such as prepending key names with a null character, which will cause an error and/or be ignored when read via [Reg](https://attack.mitre.org/software/S0075) or other utilities using the Win32 API. (Citation: Microsoft Reghide NOV 2006) Adversaries may abuse these pseudo-hidden keys to conceal payloads/commands used to maintain persistence. (Citation: TrendMicro POWELIKS AUG 2014) (Citation: SpectorOps Hiding Reg Jul 2017)

The Registry of a remote system may be modified to aid in execution of files as part of lateral movement. It requires the remote Registry service to be running on the target system. (Citation: Microsoft Remote) Often [Valid Accounts](https://attack.mitre.org/techniques/T1078) are required, along with access to the remote system's [SMB/Windows Admin Shares](https://attack.mitre.org/techniques/T1021/002) for RPC communication.



3.29 T1574.002

Used by group : APT19, menuPass, Threat Group-3390

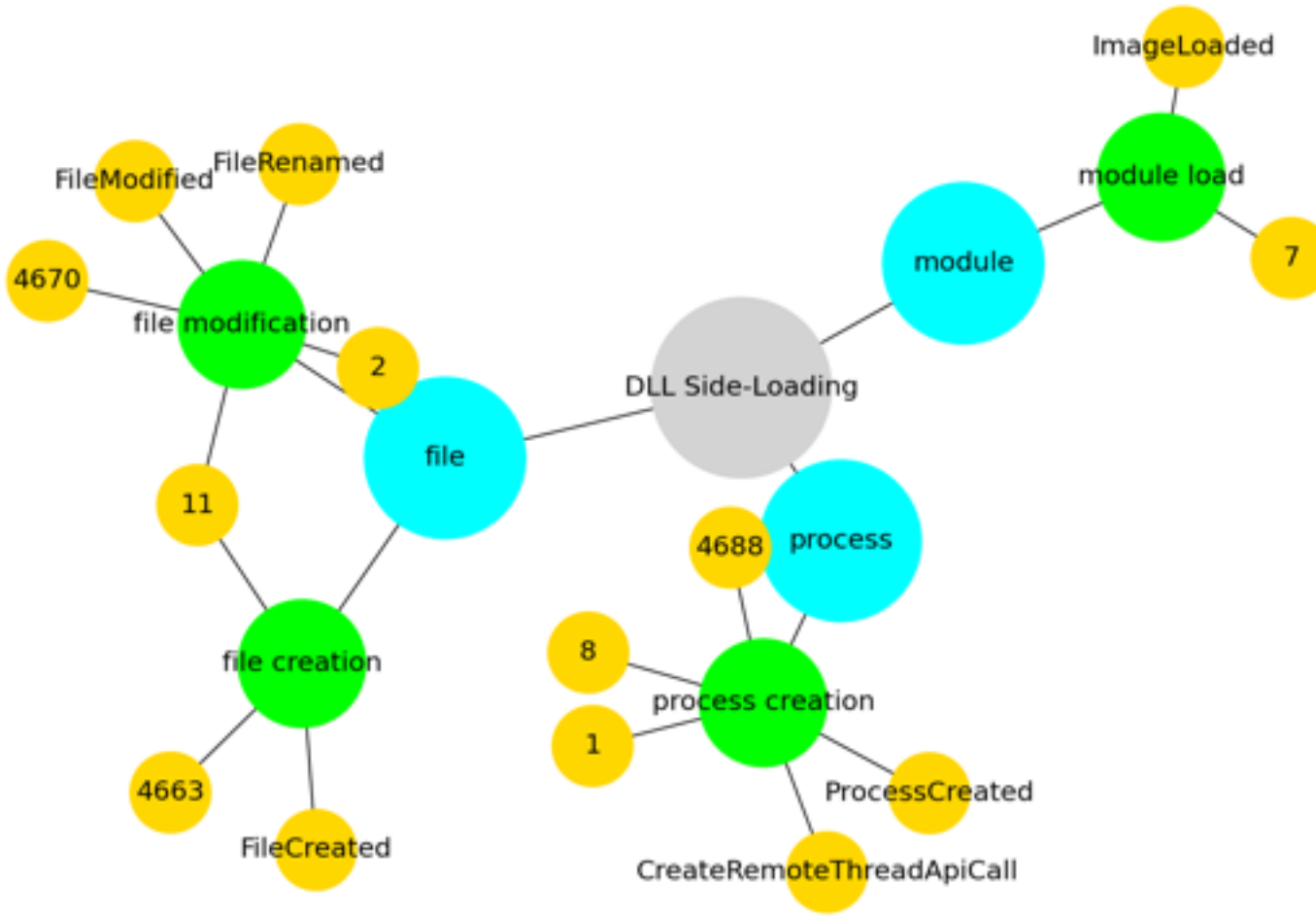
Tactic : persistence, privilege-escalation, defense-evasion

Technique : DLL Side-Loading

Adversaries may execute their own malicious payloads by side-loading DLLs. Similar to [DLL Search Order Hijacking](https://attack.mitre.org/techniques/T1574/001), side-loading involves hijacking which DLL a program loads. But rather than just planting the DLL within the search order of a program then waiting for the victim application to be invoked, adversaries may directly side-load their payloads by planting then invoking a legitimate application that executes their payload(s).

Side-loading takes advantage of the DLL search order used by the loader by positioning both the victim application and malicious payload(s) alongside each other. Adversaries likely use side-loading as a means of masking actions they perform under a legitimate, trusted, and potentially elevated system or software process. Benign executables used to side-load payloads may not be flagged

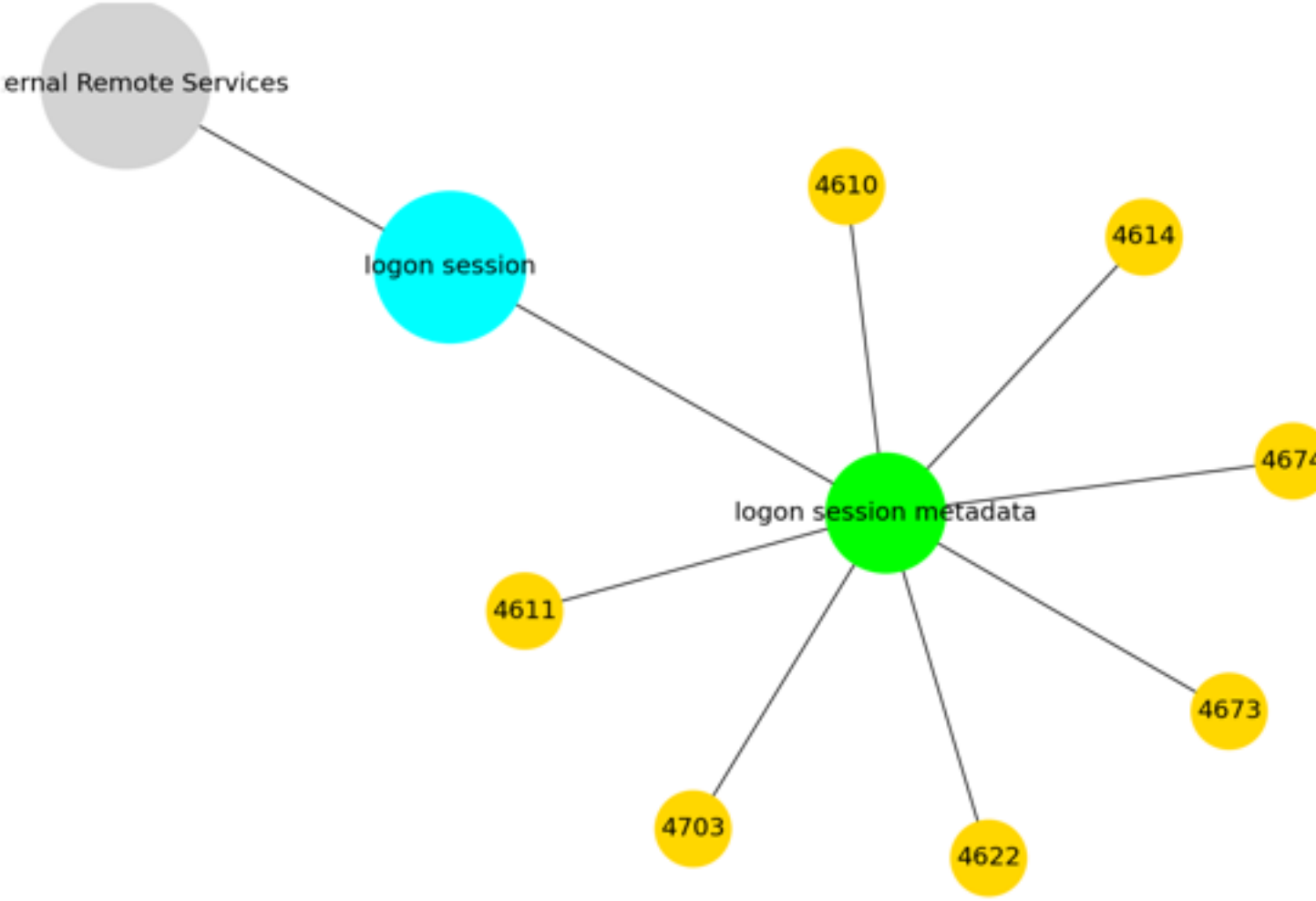
during delivery and/or execution. Adversary payloads may also be encrypted/packed or otherwise obfuscated until loaded into the memory of the trusted process.(Citation: FireEye DLL Side-Loading)



3.30 T1133

3.30 T1133

< To be corrected or added in future releases >


ernal Remote Services
logon session
4610
4614
4674
logon session metadata
4611
4673
4703
4622

4. Annexes

< To be corrected or added in future releases >

4.1 List of all techniques used

technique_id	tactic	technique	group
T1105	command-and-control	Ingress Tool Transfer	Tonto Team, Operation Wocao, Sharpshooter, APT33, OilRig, menuPass, Threat Group-3390
T1204.002	execution	Malicious File	Tonto Team, Sharpshooter, APT19, APT33, OilRig, menuPass, Threat Group-3390
T1059.001	execution	PowerShell	Tonto Team, Operation Wocao, APT19, APT33, OilRig, menuPass, Threat Group-3390
T1566.001	initial-access	Spearphishing Attachment	Tonto Team, Sharpshooter, APT19, APT33, OilRig, menuPass, Threat Group-3390
T1027	defense-evasion	Obfuscated Files or Information	Operation Wocao, APT19, APT33, OilRig, menuPass, Threat Group-3390
T1016	discovery	System Network Configuration Discovery	Operation Wocao, APT19, OilRig, menuPass, Threat Group-3390
T1056.001	collection, credential-access	Keylogging	Tonto Team, Operation Wocao, OilRig, menuPass, Threat Group-3390
T1078	defense-evasion, persistence, privilege-escalation, initial-access	Valid Accounts	Operation Wocao, APT33, OilRig, menuPass, Threat Group-3390
T1003.004	credential-access	LSA Secrets	APT33, OilRig, menuPass, Threat Group-3390
T1071.001	command-and-control	Web Protocols	APT19, APT33, OilRig, Threat Group-3390
T1119	collection	Automated Collection	Operation Wocao, OilRig, menuPass, Threat Group-3390
T1053.005	execution, persistence, privilege-escalation	Scheduled Task	Operation Wocao, APT33, OilRig, menuPass
T1059.005	execution	Visual Basic	Operation Wocao, Sharpshooter, APT33, OilRig
T1059.003	execution	Windows Command Shell	Operation Wocao, OilRig, menuPass, Threat Group-3390
T1140	defense-evasion	Deobfuscate/Decode Files or Information	APT19, OilRig, menuPass, Threat Group-3390
T1547.001	persistence, privilege-escalation	Registry Run Keys / Startup Folder	Sharpshooter, APT19, APT33, Threat Group-3390
T1588.002	resource-development	Tool	APT19, APT33, menuPass, Threat Group-3390
T1070.004	defense-evasion	File Deletion	Operation Wocao, OilRig, menuPass, Threat Group-3390
T1003.001	credential-access	LSASS Memory	Operation Wocao, APT33, OilRig, Threat Group-3390
T1049	discovery	System Network Connections Discovery	Operation Wocao, OilRig, menuPass, Threat Group-3390
T1047	execution	Windows Management Instrumentation	Operation Wocao, OilRig, menuPass, Threat Group-3390
T1046	discovery	Network Service Discovery	Operation Wocao, OilRig, menuPass, Threat Group-3390
T1505.003	persistence	Web Shell	Tonto Team, Operation Wocao, OilRig, Threat Group-3390
T1033	discovery	System Owner/User Discovery	Operation Wocao, APT19, OilRig, Threat Group-3390
T1005	collection	Data from Local System	Operation Wocao, menuPass, Threat Group-3390
T1106	execution	Native API	Operation Wocao, Sharpshooter, menuPass
T1574.001	persistence, privilege-escalation, defense-evasion	DLL Search Order Hijacking	Tonto Team, menuPass, Threat Group-3390
T1112	defense-evasion	Modify Registry	Operation Wocao, APT19, Threat Group-3390
T1574.002	persistence, privilege-escalation, defense-evasion	DLL Side-Loading	APT19, menuPass, Threat Group-3390
T1133	persistence, initial-access	External Remote Services	Operation Wocao, OilRig, Threat Group-3390
T1190	initial-access	Exploit Public-Facing Application	Operation Wocao, menuPass, Threat Group-3390
T1012	discovery	Query Registry	Operation Wocao, OilRig, Threat Group-3390
T1082	discovery	System Information Discovery	Operation Wocao, APT19, OilRig
T1018	discovery	Remote System Discovery	Operation Wocao, menuPass, Threat Group-3390
T1074.001	collection	Local Data Staging	Operation Wocao, menuPass, Threat Group-3390
T1069.001	discovery	Local Groups	Tonto Team, Operation Wocao, OilRig
T1560.001	collection	Archive via Utility	Operation Wocao, APT33, menuPass
T1203	execution	Exploitation for Client Execution	Tonto Team, APT33, Threat Group-3390
T1087.002	discovery	Domain Account	Operation Wocao, OilRig, menuPass
T1210	lateral-movement	Exploitation of Remote Services	Tonto Team, menuPass, Threat Group-3390
T1068	privilege-escalation	Exploitation for Privilege Escalation	Tonto Team, APT33, Threat Group-3390
T1021.004	lateral-movement	SSH	OilRig, menuPass

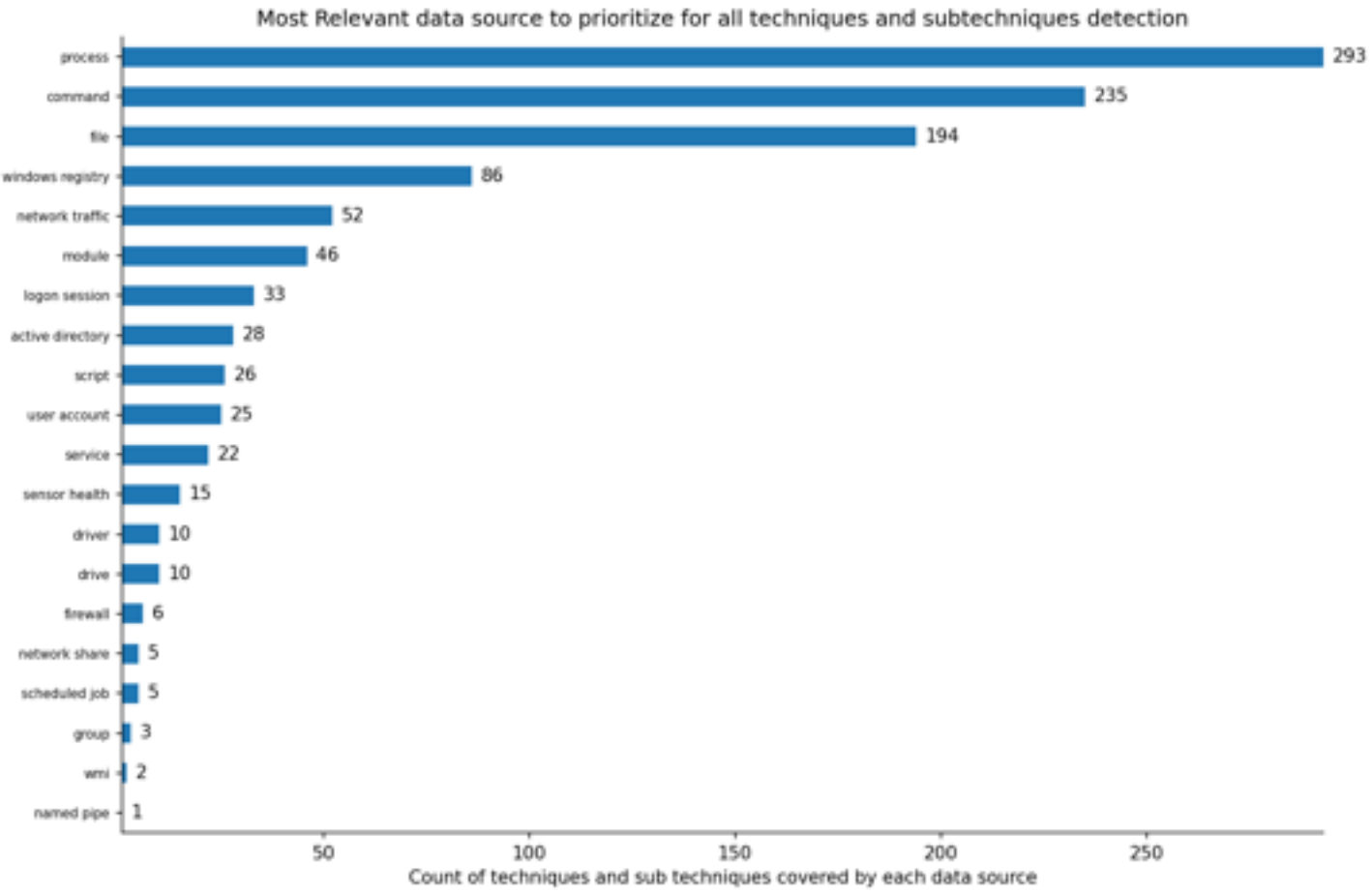
T1199	initial-access	Trusted Relationship	menuPass, Threat Group-3390
T1055.012	defense-evasion, privilege-escalation	Process Hollowing	menuPass, Threat Group-3390
T1007	discovery	System Service Discovery	Operation Wocao, OilRig
T1003.002	credential-access	Security Account Manager	menuPass, Threat Group-3390
T1074.002	collection	Remote Data Staging	menuPass, Threat Group-3390
T1036	defense-evasion	Masquerading	OilRig, menuPass
T1021.001	lateral-movement	Remote Desktop Protocol	OilRig, menuPass
T1204.001	execution	Malicious Link	APT33, OilRig
T1087.001	discovery	Local Account	OilRig, Threat Group-3390
T1048.003	exfiltration	Exfiltration Over Unencrypted Non-C2 Protocol	APT33, OilRig
T0865	initial-access-ics	Spearphishing Attachment	APT33, OilRig
T0853	execution-ics	Scripting	APT33, OilRig
T1552.001	credential-access	Credentials In Files	APT33, OilRig
T1555	credential-access	Credentials from Password Stores	APT33, OilRig
T1543.003	persistence, privilege-escalation	Windows Service	APT19, Threat Group-3390
T1189	initial-access	Drive-by Compromise	APT19, Threat Group-3390
T1555.003	credential-access	Credentials from Web Browsers	APT33, OilRig
T1003.005	credential-access	Cached Domain Credentials	APT33, OilRig
T1059	execution	Command and Scripting Interpreter	APT19, OilRig
T1566.002	initial-access	Spearphishing Link	APT33, OilRig
T1090.002	command-and-control	External Proxy	Tonto Team, menuPass
T1132.001	command-and-control	Standard Encoding	APT19, APT33
T1057	discovery	Process Discovery	Operation Wocao, OilRig
T1055	defense-evasion, privilege-escalation	Process Injection	Operation Wocao, Sharpshooter
T1083	discovery	File and Directory Discovery	Operation Wocao, menuPass
T1573.002	command-and-control	Asymmetric Cryptography	Operation Wocao, OilRig
T1135	discovery	Network Share Discovery	Tonto Team, Operation Wocao
T1027.005	defense-evasion	Indicator Removal from Tools	Operation Wocao, OilRig
T1059.006	execution	Python	Tonto Team, Operation Wocao
T1555.005	credential-access	Password Managers	Operation Wocao, Threat Group-3390
T1120	discovery	Peripheral Device Discovery	Operation Wocao, OilRig
T1041	exfiltration	Exfiltration Over C2 Channel	Operation Wocao
T1583.001	resource-development	Domains	menuPass
T1003.003	credential-access	NTDS	menuPass
T1560	collection	Archive Collected Data	menuPass
T1568.001	command-and-control	Fast Flux DNS	menuPass
T0859	persistence-ics, lateral-movement-ics	Valid Accounts	OilRig
T0869	command-and-control-ics	Standard Application Layer Protocol	OilRig
T1036.005	defense-evasion	Match Legitimate Name or Location	menuPass
T0817	initial-access-ics	Drive-by Compromise	OilRig
T1008	command-and-control	Fallback Channels	OilRig
T1110	credential-access	Brute Force	OilRig
T1113	collection	Screen Capture	OilRig
T1201	discovery	Password Policy Discovery	OilRig
T1137.004	persistence	Outlook Home Page	OilRig
T1003	credential-access	OS Credential Dumping	Tonto Team
T1070.003	defense-evasion	Clear Command History	menuPass
T1036.003	defense-evasion	Rename System Utilities	menuPass
T1562.002	defense-evasion	Disable Windows Event Logging	Threat Group-3390
T1053.002	execution, persistence, privilege-escalation	At	Threat Group-3390

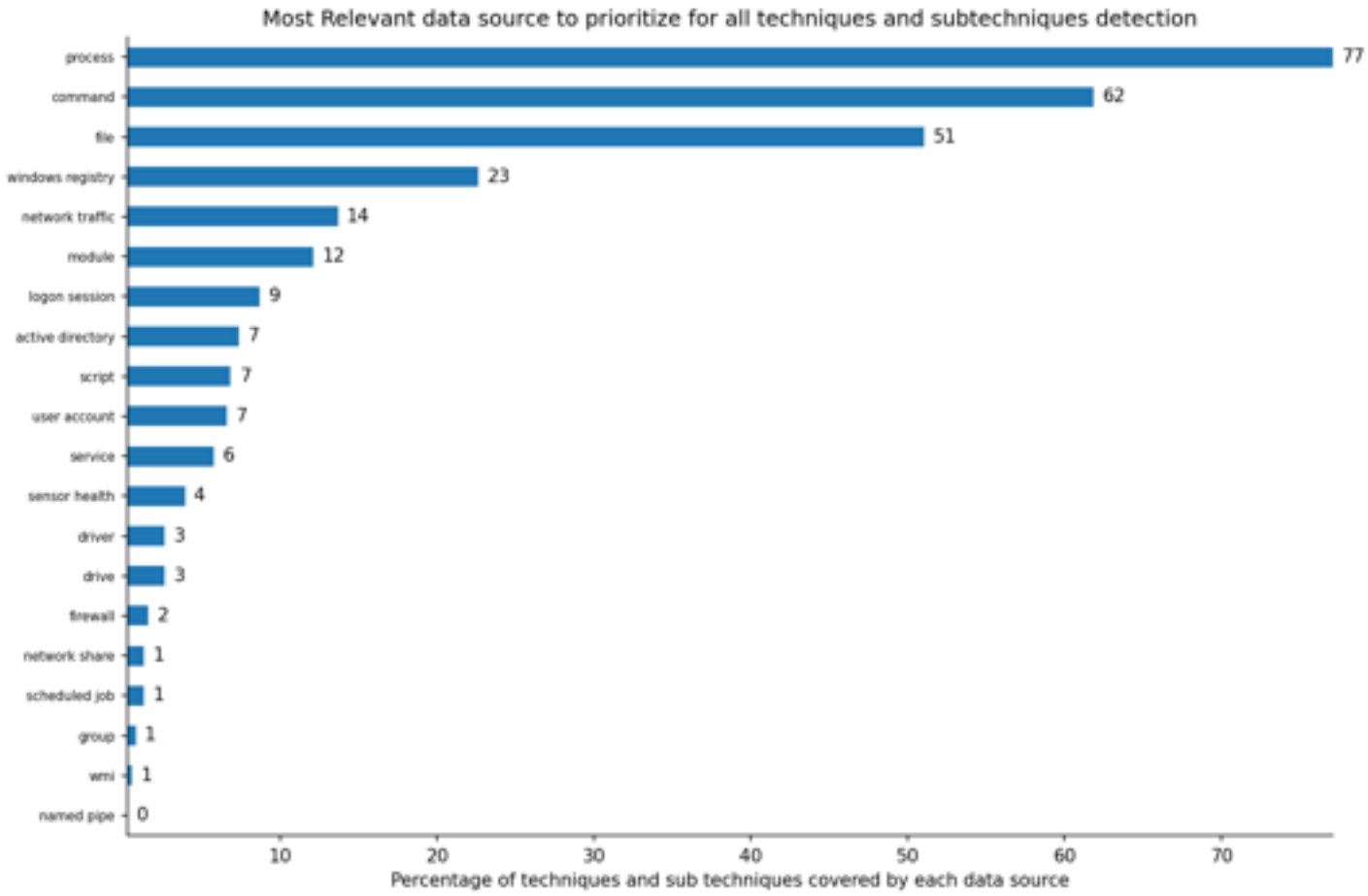
T1548.002	privilege-escalation, defense-evasion	Bypass User Account Control	Threat Group-3390
T1070.005	defense-evasion	Network Share Connection Removal	Threat Group-3390
T1027.002	defense-evasion	Software Packing	Threat Group-3390
T1021.006	lateral-movement	Windows Remote Management	Threat Group-3390
T1560.002	collection	Archive via Library	Threat Group-3390
T1567.002	exfiltration	Exfiltration to Cloud Storage	Threat Group-3390
T1553.002	defense-evasion	Code Signing	menuPass
T1195.002	initial-access	Compromise Software Supply Chain	Threat Group-3390
T1608.001	resource-development	Upload Malware	Threat Group-3390
T1608.002	resource-development	Upload Tool	Threat Group-3390
T1608.004	resource-development	Drive-by Target	Threat Group-3390
T1039	collection	Data from Network Shared Drive	menuPass
T1218.004	defense-evasion	InstallUtil	menuPass
T1218.001	defense-evasion	Compiled HTML File	OilRig
T1069.002	discovery	Domain Groups	OilRig
T1090.003	command-and-control	Multi-hop Proxy	Operation Wocao
T1218.011	defense-evasion	Rundll32	APT19
T1078.004	defense-evasion, persistence, privilege-escalation, initial-access	Cloud Accounts	APT33
T1571	command-and-control	Non-Standard Port	APT33
T1552.004	credential-access	Private Keys	Operation Wocao
T1573.001	command-and-control	Symmetric Cryptography	APT33
T1070.001	defense-evasion	Clear Windows Event Logs	Operation Wocao
T1518	discovery	Software Discovery	Operation Wocao
T1124	discovery	System Time Discovery	Operation Wocao
T1218.010	defense-evasion	Regsvr32	APT19
T1090.001	command-and-control	Internal Proxy	Operation Wocao
T1115	collection	Clipboard Data	Operation Wocao
T1564.003	defense-evasion	Hidden Window	APT19
T1111	credential-access	Multi-Factor Authentication Interception	Operation Wocao
T1095	command-and-control	Non-Application Layer Protocol	Operation Wocao
T1090	command-and-control	Proxy	Operation Wocao
T1559.002	execution	Dynamic Data Exchange	Sharpshooter
T1001	command-and-control	Data Obfuscation	Operation Wocao
T1021.002	lateral-movement	SMB/Windows Admin Shares	Operation Wocao
T1558.003	credential-access	Kerberoasting	Operation Wocao
T1003.006	credential-access	DCSync	Operation Wocao
T1562.004	defense-evasion	Disable or Modify System Firewall	Operation Wocao
T1566.003	initial-access	Spearphishing via Service	OilRig
T1497.001	defense-evasion, discovery	System Checks	OilRig
T1572	command-and-control	Protocol Tunneling	OilRig
T1071.004	command-and-control	DNS	OilRig
T1555.004	credential-access	Windows Credential Manager	OilRig
T1078.003	defense-evasion, persistence, privilege-escalation, initial-access	Local Accounts	Operation Wocao
T0852	collection-ics	Screen Capture	APT33
T1078.002	defense-evasion, persistence, privilege-escalation, initial-access	Domain Accounts	Operation Wocao
T1040	credential-access, discovery	Network Sniffing	APT33
T1546.003	privilege-escalation, persistence	Windows Management Instrumentation Event Subscription	APT33
T1570	lateral-movement	Lateral Tool Transfer	Operation Wocao
T1110.003	credential-access	Password Spraying	APT33
T1552.006	credential-access	Group Policy Preferences	APT33

T1569.002	execution	Service Execution	Operation Wocao
T1518.001	discovery	Security Software Discovery	Operation Wocao
T1030	exfiltration	Data Transfer Size Limits	Threat Group-3390

4.2 Data sources reference for covering all mitre technique

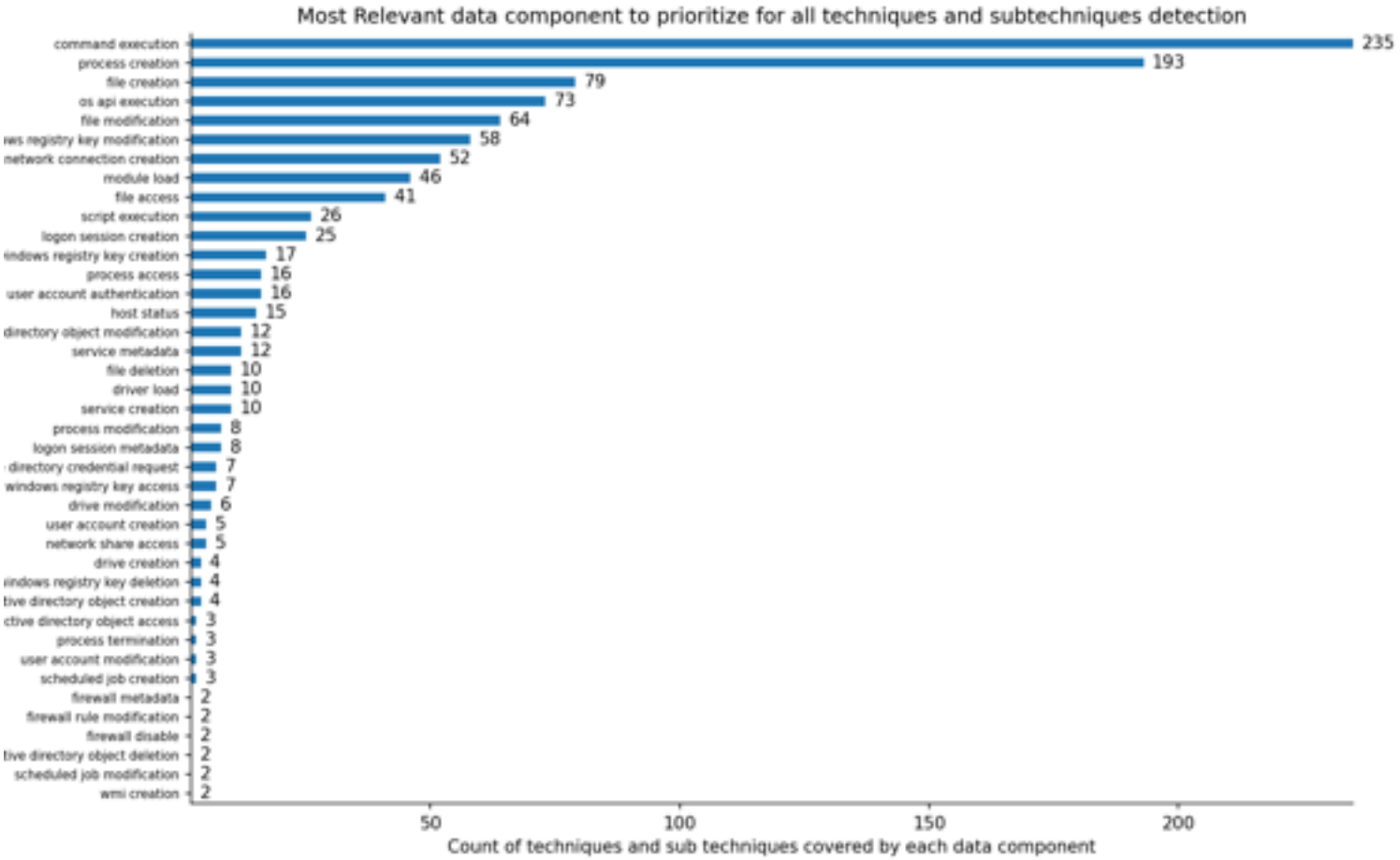
< To be corrected or added in future releases >

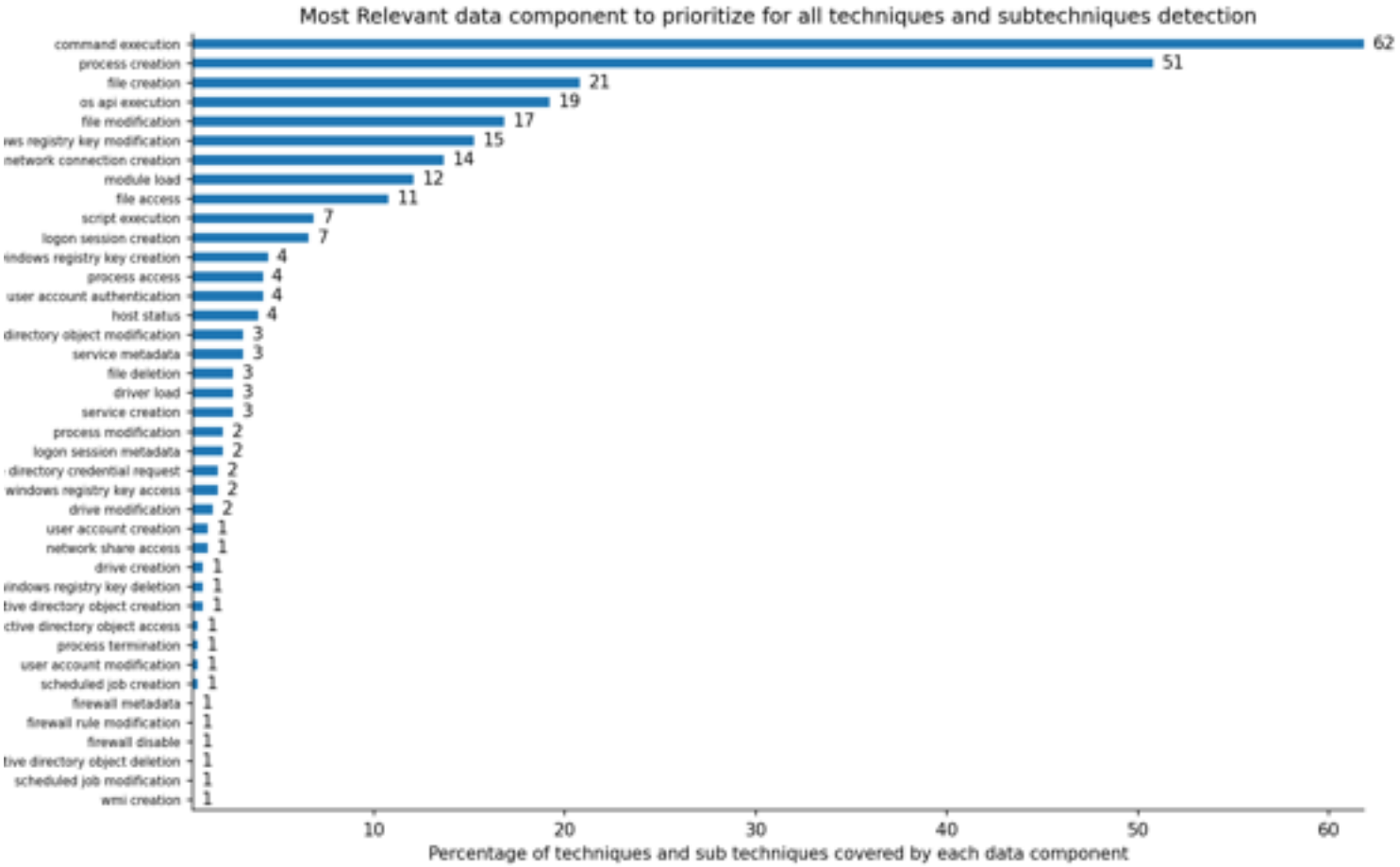




4.3 Data component reference for covering all mitre technique

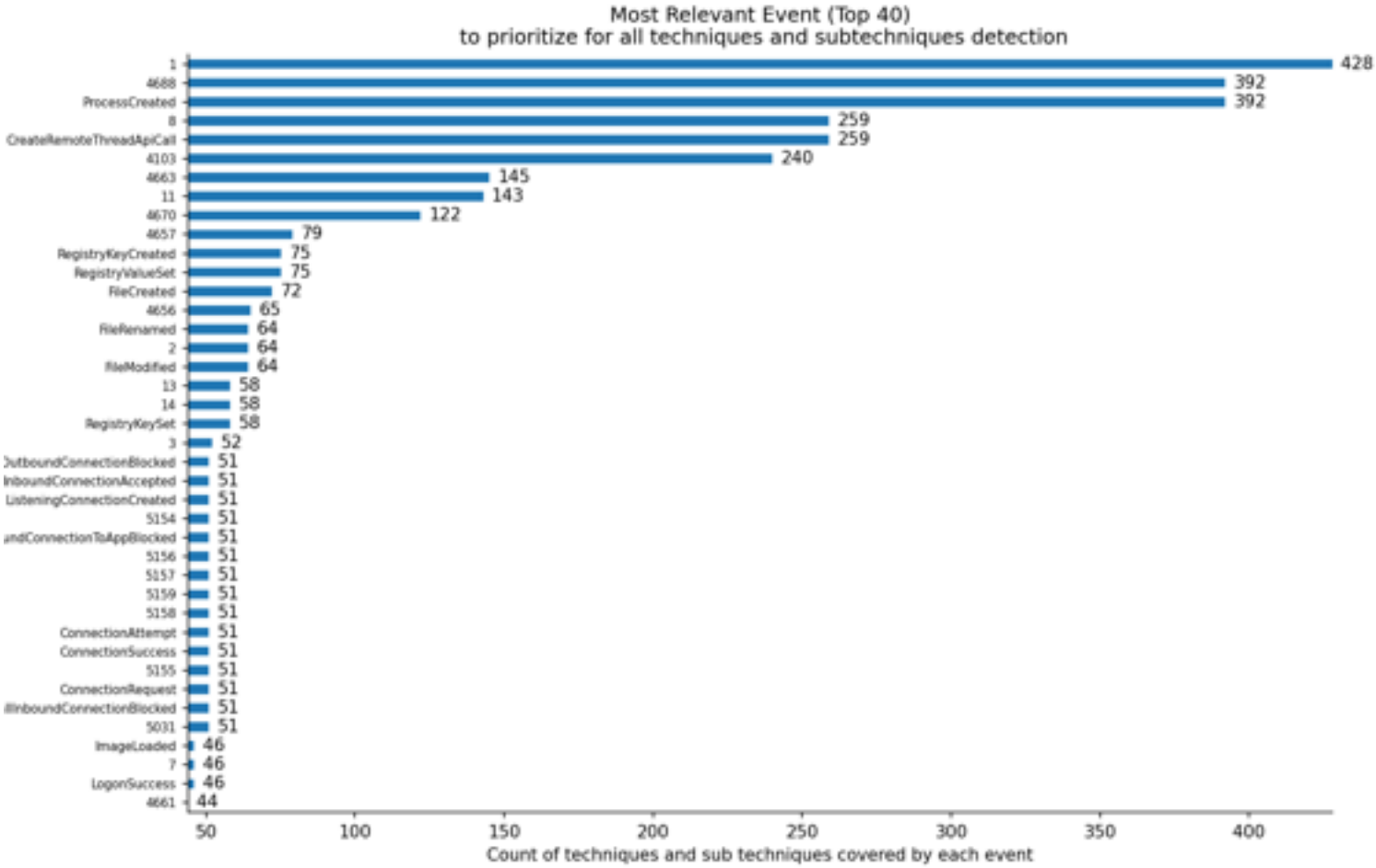
< To be corrected or added in future releases >

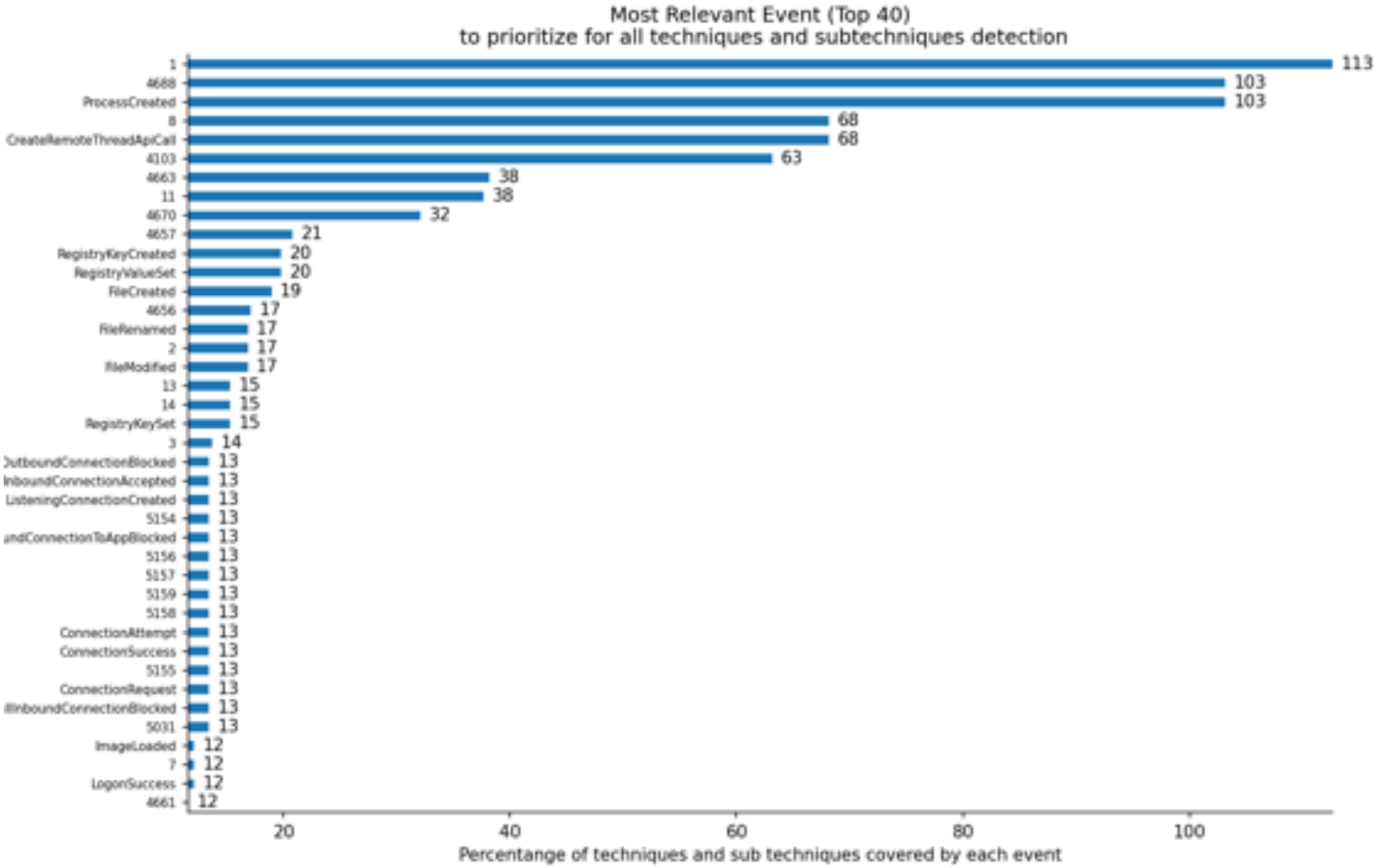

Most Relevant data component to prioritize for all techniques and subtechniques detection
command execution 235
process creation 193
file creation 79
os api execution 73
file modification 64
ws registry key modification 58
network connection creation 52
module load 46
file access 41
script execution 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
llInboundConnectionBlocked
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
llInboundConnectionBlocked
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