Tactic: Reconnaissance
ID: TA0043
Description: The adversary is trying to gather information they can use to plan future operations.

Reconnaissance consists of techniques that involve adversaries actively or passively gathering information that can be used to support targeting. Such information may include details of the victim organization, infrastructure, or staff/personnel. This information can be leveraged by the adversary to aid in other phases of the adversary lifecycle, such as using gathered information to plan and execute Initial Access, to scope and prioritize post-compromise objectives, or to drive and lead further Reconnaissance efforts.

Techniques Under the Reconnaissance Tactic

ID	NAME	DESCRIPTION
T1595	Active Scanning	Adversaries may execute active reconnaissance scans to gather information that can be used during targeting. Active scans are those where the adversary probes victim infrastructure via network traffic, as opposed to other forms of reconnaissance that do not involve direct interaction.
T1595.001	Scanning IP Blocks	Adversaries may scan victim IP blocks to gather information that can be used during targeting. Public IP addresses may be allocated to organizations by block, or a range of sequential addresses.
T1595.002	Vulnerability Scanning	Adversaries may scan victims for vulnerabilities that can be used during targeting. Vulnerability scans typically check if the configuration of a target host/application (ex: software and version) potentially aligns with the target of a specific exploit the adversary may seek to use.
T1595.003	Wordlist Scanning	Adversaries may iteratively probe infrastructure using brute-forcing and crawling techniques. While this technique employs similar methods to Brute Force, its goal is the identification of content and infrastructure rather than the discovery of valid credentials. Wordlists used in these scans may contain generic, commonly used names and file extensions or terms specific to a particular software. Adversaries may also create custom, target-specific wordlists using data gathered from other Reconnaissance techniques (ex: Gather Victim Org Information, or Search Victim-Owned Websites).

T1592	Gather Victim Host Information	Adversaries may gather information about the victim's hosts that can be used during targeting. Information about hosts may include a variety of details, including administrative data (ex: name, assigned IP, functionality, etc.) as well as specifics regarding its configuration (ex: operating system, language, etc.).
T1592.001	Hardware	Adversaries may gather information about the victim's host hardware that can be used during targeting. Information about hardware infrastructure may include a variety of details such as types and versions on specific hosts, as well as the presence of additional components that might be indicative of added defensive protections (ex: card/biometric readers, dedicated encryption hardware, etc.).
T1592.002	Software	Adversaries may gather information about the victim's host software that can be used during targeting. Information about installed software may include a variety of details such as types and versions on specific hosts, as well as the presence of additional components that might be indicative of added defensive protections (ex: antivirus, SIEMs, etc.).
T1592.003	Firmware	Adversaries may gather information about the victim's host firmware that can be used during targeting. Information about host firmware may include a variety of details such as type and versions on specific hosts, which may be used to infer more information about hosts in the environment (ex: configuration, purpose, age/patch level, etc.).
T1592.004	Client Configurations	Adversaries may gather information about the victim's client configurations that can be used during targeting. Information about client configurations may include a variety of details and settings, including operating system/version, virtualization, architecture (ex: 32 or 64 bit), language, and/or time zone.
T1589	Gather Victim Identity Information	Adversaries may gather information about the victim's identity that can be used during targeting. Information about identities may include a variety of details, including personal data (ex: employee names, email addresses, security question responses, etc.) as well as sensitive details such as credentials or multi-factor authentication (MFA) configurations.
T1589.001	Credentials	Adversaries may gather credentials that can be used during targeting. Account credentials gathered by adversaries may be those directly associated with the target victim organization or attempt to take advantage of the tendency for users to use the same passwords across personal and business accounts.

T1589.002	Email Addresses	Adversaries may gather email addresses that can be used during targeting. Even if internal instances exist, organizations may have public-facing email infrastructure and addresses for employees.
T1589.003	Employee Names	Adversaries may gather employee names that can be used during targeting. Employee names be used to derive email addresses as well as to help guide other reconnaissance efforts and/or craft more-believable lures.
T1590	Gather Victim Network Information	Adversaries may gather information about the victim's networks that can be used during targeting. Information about networks may include a variety of details, including administrative data (ex: IP ranges, domain names, etc.) as well as specifics regarding its topology and operations.
T1590.001	Domain Properties	Adversaries may gather information about the victim's network domain(s) that can be used during targeting. Information about domains and their properties may include a variety of details, including what domain(s) the victim owns as well as administrative data (ex: name, registrar, etc.) and more directly actionable information such as contacts (email addresses and phone numbers), business addresses, and name servers.
T1590.002	DNS	Adversaries may gather information about the victim's DNS that can be used during targeting. DNS information may include a variety of details, including registered name servers as well as records that outline addressing for a target’s subdomains, mail servers, and other hosts. DNS MX, TXT, and SPF records may also reveal the use of third party cloud and SaaS providers, such as Office 365, G Suite, Salesforce, or Zendesk.
T1590.003	Network Trust Dependencies	Adversaries may gather information about the victim's network trust dependencies that can be used during targeting. Information about network trusts may include a variety of details, including second or third-party organizations/domains (ex: managed service providers, contractors, etc.) that have connected (and potentially elevated) network access.
T1590.004	Network Topology	Adversaries may gather information about the victim's network topology that can be used during targeting. Information about network topologies may include a variety of details, including the physical and/or logical arrangement of both external-facing and internal network

		environments. This information may also include specifics regarding network devices (gateways, routers, etc.) and other infrastructure.
T1590.005	IP Addresses	Adversaries may gather the victim's IP addresses that can be used during targeting. Public IP addresses may be allocated to organizations by block, or a range of sequential addresses. Information about assigned IP addresses may include a variety of details, such as which IP addresses are in use. IP addresses may also enable an adversary to derive other details about a victim, such as organizational size, physical location(s), Internet service provider, and or where/how their publicly-facing infrastructure is hosted.
T1590.006	Network Security Appliances	Adversaries may gather information about the victim's network security appliances that can be used during targeting. Information about network security appliances may include a variety of details, such as the existence and specifics of deployed firewalls, content filters, and proxies/bastion hosts. Adversaries may also target information about victim network-based intrusion detection systems (NIDS) or other appliances related to defensive cybersecurity operations.
T1591	Gather Victim Org Information	Adversaries may gather information about the victim's organization that can be used during targeting. Information about an organization may include a variety of details, including the names of divisions/departments, specifics of business operations, as well as the roles and responsibilities of key employees.
T1591.001	Determine Physical Locations	Adversaries may gather the victim's physical location(s) that can be used during targeting. Information about physical locations of a target organization may include a variety of details, including where key resources and infrastructure are housed. Physical locations may also indicate what legal jurisdiction and/or authorities the victim operates within.
T1591.002	Business Relationships	Adversaries may gather information about the victim's business relationships that can be used during targeting. Information about an organization’s business relationships may include a variety of details, including second or third-party organizations/domains (ex: managed service providers, contractors, etc.) that have connected (and potentially elevated) network access. This information may also reveal supply chains and shipment paths for the victim’s hardware and software resources.

T1591.003	Identify Business Tempo	Adversaries may gather information about the victim's business tempo that can be used during targeting. Information about an organization’s business tempo may include a variety of details, including operational hours/days of the week. This information may also reveal times/dates of purchases and shipments of the victim’s hardware and software resources.
T1591.004	Identify Roles	Adversaries may gather information about identities and roles within the victim organization that can be used during targeting. Information about business roles may reveal a variety of targetable details, including identifiable information for key personnel as well as what data/resources they have access to.
T1598	Phishing for Information	Adversaries may send phishing messages to elicit sensitive information that can be used during targeting. Phishing for information is an attempt to trick targets into divulging information, frequently credentials or other actionable information. Phishing for information is different from Phishing in that the objective is gathering data from the victim rather than executing malicious code.
T1598.001	Spearphishing Service	Adversaries may send spearphishing messages via third-party services to elicit sensitive information that can be used during targeting. Spearphishing for information is an attempt to trick targets into divulging information, frequently credentials or other actionable information. Spearphishing for information frequently involves social engineering techniques, such as posing as a source with a reason to collect information (ex: Establish Accounts or Compromise Accounts) and/or sending multiple, seemingly urgent messages.
T1598.002	Spearphishing Attachment	Adversaries may send spearphishing messages with a malicious attachment to elicit sensitive information that can be used during targeting. Spearphishing for information is an attempt to trick targets into divulging information, frequently credentials or other actionable information. Spearphishing for information frequently involves social engineering techniques, such as posing as a source with a reason to collect information (ex: Establish Accounts or Compromise Accounts) and/or sending multiple, seemingly urgent messages.
T1598.003	Spearphishing Link	Adversaries may send spearphishing messages with a malicious link to elicit sensitive information that can be used during targeting. Spearphishing for information is an attempt to trick targets into divulging information, frequently credentials or other actionable information. Spearphishing for information frequently involves social engineering techniques, such as

		posing as a source with a reason to collect information (ex: Establish Accounts or Compromise Accounts) and/or sending multiple, seemingly urgent messages.
T1598.004	Spearphishing Voice	Adversaries may use voice communications to elicit sensitive information that can be used during targeting. Spearphishing for information is an attempt to trick targets into divulging information, frequently credentials or other actionable information. Spearphishing for information frequently involves social engineering techniques, such as posing as a source with a reason to collect information (ex: Impersonation) and/or creating a sense of urgency or alarm for the recipient.
T1597	Search Closed Sources	Adversaries may search and gather information about victims from closed (e.g., paid, private, or otherwise not freely available) sources that can be used during targeting. Information about victims may be available for purchase from reputable private sources and databases, such as paid subscriptions to feeds of technical/threat intelligence data. Adversaries may also purchase information from less-reputable sources such as dark web or cybercrime blackmarkets.
T1597.001	Threat Intel Vendors	Adversaries may search private data from threat intelligence vendors for information that can be used during targeting. Threat intelligence vendors may offer paid feeds or portals that offer more data than what is publicly reported. Although sensitive details (such as customer names and other identifiers) may be redacted, this information may contain trends regarding breaches such as target industries, attribution claims, and successful TTPs/countermeasures.
T1597.002	Purchase Technical Data	Adversaries may purchase technical information about victims that can be used during targeting. Information about victims may be available for purchase within reputable private sources and databases, such as paid subscriptions to feeds of scan databases or other data aggregation services. Adversaries may also purchase information from less-reputable sources such as dark web or cybercrime blackmarkets.
T1596	Search Open Technical Databases	Adversaries may search freely available technical databases for information about victims that can be used during targeting. Information about victims may be available in online databases and repositories, such as registrations of domains/certificates as well as public collections of network data/artifacts gathered from traffic and/or scans.
T1596.001	DNS/Passive DNS	Adversaries may search DNS data for information about victims that can be used during targeting. DNS information may include a variety of details, including registered name servers

		as well as records that outline addressing for a target’s subdomains, mail servers, and other hosts.
T1596.002	WHOIS	Adversaries may search public WHOIS data for information about victims that can be used during targeting. WHOIS data is stored by regional Internet registries (RIR) responsible for allocating and assigning Internet resources such as domain names. Anyone can query WHOIS servers for information about a registered domain, such as assigned IP blocks, contact information, and DNS nameservers.
T1596.003	Digital Certificates	Adversaries may search public digital certificate data for information about victims that can be used during targeting. Digital certificates are issued by a certificate authority (CA) in order to cryptographically verify the origin of signed content. These certificates, such as those used for encrypted web traffic (HTTPS SSL/TLS communications), contain information about the registered organization such as name and location.
T1596.004	CDNs	Adversaries may search content delivery network (CDN) data about victims that can be used during targeting. CDNs allow an organization to host content from a distributed, load balanced array of servers. CDNs may also allow organizations to customize content delivery based on the requestor’s geographical region.
T1596.005	Scan Databases	Adversaries may search within public scan databases for information about victims that can be used during targeting. Various online services continuously publish the results of Internet scans/surveys, often harvesting information such as active IP addresses, hostnames, open ports, certificates, and even server banners.
T1593	Search Open Websites/Domains	Adversaries may search freely available websites and/or domains for information about victims that can be used during targeting. Information about victims may be available in various online sites, such as social media, new sites, or those hosting information about business operations such as hiring or requested/rewarded contracts.
T1593.001	Social Media	Adversaries may search social media for information about victims that can be used during targeting. Social media sites may contain various information about a victim organization, such as business announcements as well as information about the roles, locations, and interests of staff.

T1593.002	Search Engines	Adversaries may use search engines to collect information about victims that can be used during targeting. Search engine services typical crawl online sites to index context and may provide users with specialized syntax to search for specific keywords or specific types of content (i.e. filetypes).
T1593.003	Code Repositories	Adversaries may search public code repositories for information about victims that can be used during targeting. Victims may store code in repositories on various third-party websites such as GitHub, GitLab, SourceForge, and BitBucket. Users typically interact with code repositories through a web application or command-line utilities such as git.
T1594	Search Victim-Owned Websites	Adversaries may search websites owned by the victim for information that can be used during targeting. Victim-owned websites may contain a variety of details, including names of departments/divisions, physical locations, and data about key employees such as names, roles, and contact info (ex: Email Addresses). These sites may also have details highlighting business operations and relationships.

Tactic: Resource Development
ID: TA0042
Description:
The adversary is trying to establish resources they can use to support operations.

Resource Development consists of techniques that involve adversaries creating, purchasing, or compromising/stealing resources that can be used to support targeting. Such resources include infrastructure, accounts, or capabilities. These resources can be leveraged by the adversary to aid in other phases of the adversary lifecycle, such as using purchased domains to support Command and Control, email accounts for phishing as a part of Initial Access, or stealing code signing certificates to help with Defense Evasion.

Techniques Under the Resource Development Tactic

ID	NAME	DESCRIPTION

T1650	Acquire Access	Adversaries may purchase or otherwise acquire an existing access to a target system or network. A variety of online services and initial access broker networks are available to sell access to previously compromised systems. In some cases, adversary groups may form partnerships to share compromised systems with each other.
T1583	Acquire Infrastructure	Adversaries may buy, lease, rent, or obtain infrastructure that can be used during targeting. A wide variety of infrastructure exists for hosting and orchestrating adversary operations. Infrastructure solutions include physical or cloud servers, domains, and third-party web services. Some infrastructure providers offer free trial periods, enabling infrastructure acquisition at limited to no cost. Additionally, botnets are available for rent or purchase.
T1583.001	Domains	Adversaries may acquire domains that can be used during targeting. Domain names are the human readable names used to represent one or more IP addresses. They can be purchased or, in some cases, acquired for free.
T1583.002	DNS Server	Adversaries may set up their own Domain Name System (DNS) servers that can be used during targeting. During post-compromise activity, adversaries may utilize DNS traffic for various tasks, including for Command and Control (ex: Application Layer Protocol). Instead of hijacking existing DNS servers, adversaries may opt to configure and run their own DNS servers in support of operations.
T1583.003	Virtual Private Server	Adversaries may rent Virtual Private Servers (VPSs) that can be used during targeting. There exist a variety of cloud service providers that will sell virtual machines/containers as a service. By utilizing a VPS, adversaries can make it difficult to physically tie back operations to them. The use of cloud infrastructure can also make it easier for adversaries to rapidly provision, modify, and shut down their infrastructure.
T1583.004	Server	Adversaries may buy, lease, rent, or obtain physical servers that can be used during targeting. Use of servers allows an adversary to stage, launch, and execute an operation. During post-compromise activity, adversaries may utilize servers for various tasks, such as watering hole operations in Drive-by Compromise, enabling Phishing operations, or facilitating Command and Control. Instead of compromising a third-party Server or renting a Virtual Private Server, adversaries may opt to configure and run their own servers in support of operations. Free trial periods of cloud servers may also be abused.

T1583.005	Botnet	Adversaries may buy, lease, or rent a network of compromised systems that can be used during targeting. A botnet is a network of compromised systems that can be instructed to perform coordinated tasks. Adversaries may purchase a subscription to use an existing botnet from a booter/stresser service. With a botnet at their disposal, adversaries may perform follow-on activity such as large-scale Phishing or Distributed Denial of Service (DDoS).
T1583.006	Web Services	Adversaries may register for web services that can be used during targeting. A variety of popular websites exist for adversaries to register for a web-based service that can be abused during later stages of the adversary lifecycle, such as during Command and Control (Web Service), Exfiltration Over Web Service, or Phishing. Using common services, such as those offered by Google or Twitter, makes it easier for adversaries to hide in expected noise. By utilizing a web service, adversaries can make it difficult to physically tie back operations to them.
T1583.007	Serverless	Adversaries may purchase and configure serverless cloud infrastructure, such as Cloudflare Workers, AWS Lambda functions, or Google Apps Scripts, that can be used during targeting. By utilizing serverless infrastructure, adversaries can make it more difficult to attribute infrastructure used during operations back to them.
T1583.008	Malvertising	Adversaries may purchase online advertisements that can be abused to distribute malware to victims. Ads can be purchased to plant as well as favorably position artifacts in specific locations online, such as prominently placed within search engine results. These ads may make it more difficult for users to distinguish between actual search results and advertisements. Purchased ads may also target specific audiences using the advertising network's capabilities, potentially further taking advantage of the trust inherently given to search engines and popular websites.
T1586	Compromise Accounts	Adversaries may compromise accounts with services that can be used during targeting. For operations incorporating social engineering, the utilization of an online persona may be important. Rather than creating and cultivating accounts (i.e. Establish Accounts), adversaries may compromise existing accounts. Utilizing an existing persona may engender a level of trust in a potential victim if they have a relationship, or knowledge of, the compromised persona.

T1586.001	Social Media Accounts	Adversaries may compromise social media accounts that can be used during targeting. For operations incorporating social engineering, the utilization of an online persona may be important. Rather than creating and cultivating social media profiles (i.e. Social Media Accounts), adversaries may compromise existing social media accounts. Utilizing an existing persona may engender a level of trust in a potential victim if they have a relationship, or knowledge of, the compromised persona.
T1586.002	Email Accounts	Adversaries may compromise email accounts that can be used during targeting. Adversaries can use compromised email accounts to further their operations, such as leveraging them to conduct Phishing for Information, Phishing, or large-scale spam email campaigns. Utilizing an existing persona with a compromised email account may engender a level of trust in a potential victim if they have a relationship with, or knowledge of, the compromised persona. Compromised email accounts can also be used in the acquisition of infrastructure (ex: Domains).
T1586.003	Cloud Accounts	Adversaries may compromise cloud accounts that can be used during targeting. Adversaries can use compromised cloud accounts to further their operations, including leveraging cloud storage services such as Dropbox, Microsoft OneDrive, or AWS S3 buckets for Exfiltration to Cloud Storage or to Upload Tools. Cloud accounts can also be used in the acquisition of infrastructure, such as Virtual Private Servers or Serverless infrastructure. Additionally, cloud-based messaging services such as Twilio, SendGrid, AWS End User Messaging, AWS SNS (Simple Notification Service), or AWS SES (Simple Email Service) may be leveraged for spam or Phishing. Compromising cloud accounts may allow adversaries to develop sophisticated capabilities without managing their own servers.

T1584	Compromise Infrastructure	Adversaries may compromise third-party infrastructure that can be used during targeting. Infrastructure solutions include physical or cloud servers, domains, network devices, and third-party web and DNS services. Instead of buying, leasing, or renting infrastructure an adversary may compromise infrastructure and use it during other phases of the adversary lifecycle. Additionally, adversaries may compromise numerous machines to form a botnet they can leverage.
T1584.001	Domains	Adversaries may hijack domains and/or subdomains that can be used during targeting. Domain registration hijacking is the act of changing the registration of a domain name without the permission of the original registrant. Adversaries may gain access to an email account for the person listed as the owner of the domain. The adversary can then claim that they forgot their password in order to make changes to the domain registration. Other possibilities include social engineering a domain registration help desk to gain access to an account, taking advantage of renewal process gaps, or compromising a cloud service that enables managing domains (e.g., AWS Route53).
T1584.002	DNS Server	Adversaries may compromise third-party DNS servers that can be used during targeting. During post-compromise activity, adversaries may utilize DNS traffic for various tasks, including for Command and Control (ex: Application Layer Protocol). Instead of setting up their own DNS servers, adversaries may compromise third-party DNS servers in support of operations.
T1584.003	Virtual Private Server	Adversaries may compromise third-party Virtual Private Servers (VPSs) that can be used during targeting. There exist a variety of cloud service providers that will sell virtual machines/containers as a service. Adversaries may compromise VPSs purchased by third-party entities. By compromising a VPS to use as infrastructure, adversaries can make it difficult to physically tie back operations to themselves.

T1584.004	Server	Adversaries may compromise third-party servers that can be used during targeting. Use of servers allows an adversary to stage, launch, and execute an operation. During post-compromise activity, adversaries may utilize servers for various tasks, including for Command and Control. Instead of purchasing a Server or Virtual Private Server, adversaries may compromise third-party servers in support of operations.
T1584.005	Botnet	Adversaries may compromise numerous third-party systems to form a botnet that can be used during targeting. A botnet is a network of compromised systems that can be instructed to perform coordinated tasks. Instead of purchasing/renting a botnet from a booter/stresser service, adversaries may build their own botnet by compromising numerous third-party systems. Adversaries may also conduct a takeover of an existing botnet, such as redirecting bots to adversary-controlled C2 servers. With a botnet at their disposal, adversaries may perform follow-on activity such as large-scale Phishing or Distributed Denial of Service (DDoS).
T1584.006	Web Services	Adversaries may compromise access to third-party web services that can be used during targeting. A variety of popular websites exist for legitimate users to register for web-based services, such as GitHub, Twitter, Dropbox, Google, SendGrid, etc. Adversaries may try to take ownership of a legitimate user's access to a web service and use that web service as infrastructure in support of cyber operations. Such web services can be abused during later stages of the adversary lifecycle, such as during Command and Control (Web Service), Exfiltration Over Web Service, or Phishing. Using common services, such as those offered by Google or Twitter, makes it easier for adversaries to hide in expected noise. By utilizing a web service, particularly when access is stolen from legitimate users, adversaries can make it difficult to physically tie back operations to them. Additionally, leveraging compromised web-based email services may allow adversaries to leverage the trust associated with legitimate domains.

T1584.007	Serverless	Adversaries may compromise serverless cloud infrastructure, such as Cloudflare Workers, AWS Lambda functions, or Google Apps Scripts, that can be used during targeting. By utilizing serverless infrastructure, adversaries can make it more difficult to attribute infrastructure used during operations back to them.
T1584.008	Network Devices	Adversaries may compromise third-party network devices that can be used during targeting. Network devices, such as small office/home office (SOHO) routers, may be compromised where the adversary's ultimate goal is not Initial Access to that environment -- instead leveraging these devices to support additional targeting.
T1587	Develop Capabilities	Adversaries may build capabilities that can be used during targeting. Rather than purchasing, freely downloading, or stealing capabilities, adversaries may develop their own capabilities in-house. This is the process of identifying development requirements and building solutions such as malware, exploits, and self-signed certificates. Adversaries may develop capabilities to support their operations throughout numerous phases of the adversary lifecycle.
T1587.001	Malware	Adversaries may develop malware and malware components that can be used during targeting. Building malicious software can include the development of payloads, droppers, post-compromise tools, backdoors (including backdoored images), packers, C2 protocols, and the creation of infected removable media. Adversaries may develop malware to support their operations, creating a means for maintaining control of remote machines, evading defenses, and executing post-compromise behaviors.

T1587.002	Code Signing Certificates	Adversaries may create self-signed code signing certificates that can be used during targeting. Code signing is the process of digitally signing executables and scripts to confirm the software author and guarantee that the code has not been altered or corrupted. Code signing provides a level of authenticity for a program from the developer and a guarantee that the program has not been tampered with. Users and/or security tools may trust a signed piece of code more than an unsigned piece of code even if they don't know who issued the certificate or who the author is.
T1587.003	Digital Certificates	Adversaries may create self-signed SSL/TLS certificates that can be used during targeting. SSL/TLS certificates are designed to instill trust. They include information about the key, information about its owner's identity, and the digital signature of an entity that has verified the certificate's contents are correct. If the signature is valid, and the person examining the certificate trusts the signer, then they know they can use that key to communicate with its owner. In the case of self-signing, digital certificates will lack the element of trust associated with the signature of a third-party certificate authority (CA).
T1587.004	Exploits	Adversaries may develop exploits that can be used during targeting. An exploit takes advantage of a bug or vulnerability in order to cause unintended or unanticipated behavior to occur on computer hardware or software. Rather than finding/modifying exploits from online or purchasing them from exploit vendors, an adversary may develop their own exploits. Adversaries may use information acquired via Vulnerabilities to focus exploit development efforts. As part of the exploit development process, adversaries may uncover exploitable vulnerabilities through methods such as fuzzing and patch analysis.

T1585	Establish Accounts	Adversaries may create and cultivate accounts with services that can be used during targeting. Adversaries can create accounts that can be used to build a persona to further operations. Persona development consists of the development of public information, presence, history and appropriate affiliations. This development could be applied to social media, website, or other publicly available information that could be referenced and scrutinized for legitimacy over the course of an operation using that persona or identity.
T1585.001	Social Media Accounts	Adversaries may create and cultivate social media accounts that can be used during targeting. Adversaries can create social media accounts that can be used to build a persona to further operations. Persona development consists of the development of public information, presence, history and appropriate affiliations.
T1585.002	Email Accounts	Adversaries may create email accounts that can be used during targeting. Adversaries can use accounts created with email providers to further their operations, such as leveraging them to conduct Phishing for Information or Phishing. Establishing email accounts may also allow adversaries to abuse free services – such as trial periods – to Acquire Infrastructure for follow-on purposes.
T1585.003	Cloud Accounts	Adversaries may create accounts with cloud providers that can be used during targeting. Adversaries can use cloud accounts to further their operations, including leveraging cloud storage services such as Dropbox, MEGA, Microsoft OneDrive, or AWS S3 buckets for Exfiltration to Cloud Storage or to Upload Tools. Cloud accounts can also be used in the acquisition of infrastructure, such as Virtual Private Servers or Serverless infrastructure. Establishing cloud accounts may allow adversaries to develop sophisticated capabilities without managing their own servers.

T1588	Obtain Capabilities	Adversaries may buy and/or steal capabilities that can be used during targeting. Rather than developing their own capabilities in-house, adversaries may purchase, freely download, or steal them. Activities may include the acquisition of malware, software (including licenses), exploits, certificates, and information relating to vulnerabilities. Adversaries may obtain capabilities to support their operations throughout numerous phases of the adversary lifecycle.
T1588.001	Malware	Adversaries may buy, steal, or download malware that can be used during targeting. Malicious software can include payloads, droppers, post-compromise tools, backdoors, packers, and C2 protocols. Adversaries may acquire malware to support their operations, obtaining a means for maintaining control of remote machines, evading defenses, and executing post-compromise behaviors.
T1588.002	Tool	Adversaries may buy, steal, or download software tools that can be used during targeting. Tools can be open or closed source, free or commercial. A tool can be used for malicious purposes by an adversary, but (unlike malware) were not intended to be used for those purposes (ex: PsExec). Tool acquisition can involve the procurement of commercial software licenses, including for red teaming tools such as Cobalt Strike. Commercial software may be obtained through purchase, stealing licenses (or licensed copies of the software), or cracking trial versions.
T1588.003	Code Signing Certificates	Adversaries may buy and/or steal code signing certificates that can be used during targeting. Code signing is the process of digitally signing executables and scripts to confirm the software author and guarantee that the code has not been altered or corrupted. Code signing provides a level of authenticity for a program from the developer and a guarantee that the program has not been tampered with. Users and/or security tools may trust a signed piece of code more than an unsigned piece of code even if they don't know who issued the certificate or who the author is.

T1588.004	Digital Certificates	Adversaries may buy and/or steal SSL/TLS certificates that can be used during targeting. SSL/TLS certificates are designed to instill trust. They include information about the key, information about its owner's identity, and the digital signature of an entity that has verified the certificate's contents are correct. If the signature is valid, and the person examining the certificate trusts the signer, then they know they can use that key to communicate with its owner.
T1588.005	Exploits	Adversaries may buy, steal, or download exploits that can be used during targeting. An exploit takes advantage of a bug or vulnerability in order to cause unintended or unanticipated behavior to occur on computer hardware or software. Rather than developing their own exploits, an adversary may find/modify exploits from online or purchase them from exploit vendors.
T1588.006	Vulnerabilities	Adversaries may acquire information about vulnerabilities that can be used during targeting. A vulnerability is a weakness in computer hardware or software that can, potentially, be exploited by an adversary to cause unintended or unanticipated behavior to occur. Adversaries may find vulnerability information by searching open databases or gaining access to closed vulnerability databases.
T1588.007	Artificial Intelligence	Adversaries may obtain access to generative artificial intelligence tools, such as large language models (LLMs), to aid various techniques during targeting. These tools may be used to inform, bolster, and enable a variety of malicious tasks including conducting Reconnaissance, creating basic scripts, assisting social engineering, and even developing payloads.

T1608	Stage Capabilities	Adversaries may upload, install, or otherwise set up capabilities that can be used during targeting. To support their operations, an adversary may need to take capabilities they developed (Develop Capabilities) or obtained (Obtain Capabilities) and stage them on infrastructure under their control. These capabilities may be staged on infrastructure that was previously purchased/rented by the adversary (Acquire Infrastructure) or was otherwise compromised by them (Compromise Infrastructure). Capabilities may also be staged on web services, such as GitHub or Pastebin, or on Platform-as-a-Service (PaaS) offerings that enable users to easily provision applications.
T1608.001	Upload Malware	Adversaries may upload malware to third-party or adversary controlled infrastructure to make it accessible during targeting. Malicious software can include payloads, droppers, post-compromise tools, backdoors, and a variety of other malicious content. Adversaries may upload malware to support their operations, such as making a payload available to a victim network to enable Ingress Tool Transfer by placing it on an Internet accessible web server.
T1608.002	Upload Tool	Adversaries may upload tools to third-party or adversary controlled infrastructure to make it accessible during targeting. Tools can be open or closed source, free or commercial. Tools can be used for malicious purposes by an adversary, but (unlike malware) were not intended to be used for those purposes (ex: PsExec). Adversaries may upload tools to support their operations, such as making a tool available to a victim network to enable Ingress Tool Transfer by placing it on an Internet accessible web server.

T1608.003	Install Digital Certificate	Adversaries may install SSL/TLS certificates that can be used during targeting. SSL/TLS certificates are files that can be installed on servers to enable secure communications between systems. Digital certificates include information about the key, information about its owner's identity, and the digital signature of an entity that has verified the certificate's contents are correct. If the signature is valid, and the person examining the certificate trusts the signer, then they know they can use that key to communicate securely with its owner. Certificates can be uploaded to a server, then the server can be configured to use the certificate to enable encrypted communication with it.
T1608.004	Drive-by Target	Adversaries may prepare an operational environment to infect systems that visit a website over the normal course of browsing. Endpoint systems may be compromised through browsing to adversary controlled sites, as in Drive-by Compromise. In such cases, the user's web browser is typically targeted for exploitation (often not requiring any extra user interaction once landing on the site), but adversaries may also set up websites for non-exploitation behavior such as Application Access Token. Prior to Drive-by Compromise, adversaries must stage resources needed to deliver that exploit to users who browse to an adversary controlled site. Drive-by content can be staged on adversary controlled infrastructure that has been acquired (Acquire Infrastructure) or previously compromised (Compromise Infrastructure).
T1608.005	Link Target	Adversaries may put in place resources that are referenced by a link that can be used during targeting. An adversary may rely upon a user clicking a malicious link in order to divulge information (including credentials) or to gain execution, as in Malicious Link. Links can be used for spearphishing, such as sending an email accompanied by social engineering text to coax the user to actively click or copy and paste a URL into a browser. Prior to a phish for information (as in Spearphishing Link) or a phish to gain initial access to a system (as in Spearphishing Link), an adversary must set up the resources for a link target for the spearphishing link.

T1608.006	SEO Poisoning	Adversaries may poison mechanisms that influence search engine optimization (SEO) to further lure staged capabilities towards potential victims. Search engines typically display results to users based on purchased ads as well as the site's ranking/score/reputation calculated by their web crawlers and algorithms.

Tactic: Initial Access
ID: TA0001
Description:
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.

Techniques Under the Initial Access Tactic

ID	NAME	DESCRIPTION

T1659	Content Injection	Adversaries may gain access and continuously communicate with victims by injecting malicious content into systems through online network traffic. Rather than luring victims to malicious payloads hosted on a compromised website (i.e., Drive-by Target followed by Drive-by Compromise), adversaries may initially access victims through compromised data-transfer channels where they can manipulate traffic and/or inject their own content. These compromised online network channels may also be used to deliver additional payloads (i.e., Ingress Tool Transfer) and other data to already compromised systems.
T1189	Drive-by Compromise	Adversaries may gain 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 Token.
T1190	Exploit Public-Facing Application	Adversaries may attempt to exploit a weakness in an Internet-facing host or system to initially access a network. The weakness in the system can be a software bug, a temporary glitch, or a misconfiguration.
T1133	External Remote Services	Adversaries may leverage external-facing remote services to initially access and/or persist within a network. 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 and VNC can also be used externally.

T1200	Hardware Additions	Adversaries may introduce computer accessories, networking hardware, or other computing devices into a system or network that can be used as a vector to gain access. Rather than just connecting and distributing payloads via removable storage (i.e. Replication Through Removable Media), more robust hardware additions can be used to introduce new functionalities and/or features into a system that can then be abused.
T1566	Phishing	Adversaries may send phishing messages to gain access to victim systems. All forms of phishing are electronically delivered social engineering. Phishing can be targeted, known as spearphishing. In spearphishing, a specific individual, company, or industry will be targeted by the adversary. More generally, adversaries can conduct non-targeted phishing, such as in mass malware spam campaigns.
T1566.001	Spearphishing Attachment	Adversaries may send spearphishing emails with a malicious attachment in an attempt to gain access to victim systems. Spearphishing attachment is a specific variant of spearphishing. Spearphishing attachment is different from other forms of spearphishing in that it employs the use of malware attached to an email. All forms of spearphishing are electronically delivered social engineering targeted at a specific individual, company, or industry. In this scenario, adversaries attach a file to the spearphishing email and usually rely upon User Execution to gain execution. Spearphishing may also involve social engineering techniques, such as posing as a trusted source.

T1566.002	Spearphishing Link	Adversaries may send spearphishing emails with a malicious link in an attempt to gain access to victim systems. Spearphishing with a link is a specific variant of spearphishing. It is different from other forms of spearphishing in that it employs the use of links to download malware contained in email, instead of attaching malicious files to the email itself, to avoid defenses that may inspect email attachments. Spearphishing may also involve social engineering techniques, such as posing as a trusted source.
T1566.003	Spearphishing via Service	Adversaries may send spearphishing messages via third-party services in an attempt to gain access to victim systems. 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.
T1566.004	Spearphishing Voice	Adversaries may use voice communications to ultimately gain access to victim systems. Spearphishing voice is a specific variant of spearphishing. It is different from other forms of spearphishing in that is employs the use of manipulating a user into providing access to systems through a phone call or other forms of voice communications. Spearphishing frequently involves social engineering techniques, such as posing as a trusted source (ex: Impersonation) and/or creating a sense of urgency or alarm for the recipient.
T1091	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.

T1195	Supply Chain Compromise	Adversaries may manipulate products or product delivery mechanisms prior to receipt by a final consumer for the purpose of data or system compromise.
T1195.001	Compromise Software Dependencies and Development Tools	Adversaries may manipulate software dependencies and development tools prior to receipt by a final consumer for the purpose of data or system compromise. Applications often depend on external software to function properly. Popular open source projects that are used as dependencies in many applications may be targeted as a means to add malicious code to users of the dependency.
T1195.002	Compromise Software Supply Chain	Adversaries may manipulate application software prior to receipt by a final consumer for the purpose of data or system compromise. Supply chain compromise of software can take place in a number of ways, including manipulation of the application source code, manipulation of the update/distribution mechanism for that software, or replacing compiled releases with a modified version.
T1195.003	Compromise Hardware Supply Chain	Adversaries may manipulate hardware components in products prior to receipt by a final consumer for the purpose of data or system compromise. By modifying hardware or firmware in the supply chain, adversaries can insert a backdoor into consumer networks that may be difficult to detect and give the adversary a high degree of control over the system. Hardware backdoors may be inserted into various devices, such as servers, workstations, network infrastructure, or peripherals.
T1199	Trusted Relationship	Adversaries may breach or otherwise leverage organizations who have access to intended victims. Access through trusted third party relationship abuses an existing connection that may not be protected or receives less scrutiny than standard mechanisms of gaining access to a network.

T1078	Valid Accounts	Adversaries may obtain and abuse credentials of existing accounts as a means of gaining Initial Access, Persistence, Privilege Escalation, or Defense Evasion. Compromised credentials may be used to bypass access controls placed on various resources on systems within the network and may even be used for persistent access to remote systems and externally available services, such as VPNs, Outlook Web Access, network devices, 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.
T1078.001	Default Accounts	Adversaries may obtain and abuse credentials of a default account as a means of gaining Initial Access, Persistence, Privilege Escalation, or Defense Evasion. Default accounts are those that are built-into an OS, such as the Guest or Administrator accounts on Windows systems. Default accounts also include default factory/provider set accounts on other types of systems, software, or devices, including the root user account in AWS and the default service account in Kubernetes.
T1078.002	Domain Accounts	Adversaries may obtain and abuse credentials of a domain account as a means of gaining Initial Access, Persistence, Privilege Escalation, or Defense Evasion. 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.
T1078.003	Local Accounts	Adversaries may obtain and abuse credentials of a local account as a means of gaining Initial Access, Persistence, Privilege Escalation, or Defense Evasion. Local accounts are those configured by an organization for use by users, remote support, services, or for administration on a single system or service.

T1078.004	Cloud Accounts	Valid accounts in cloud environments may allow adversaries to perform actions to achieve Initial Access, Persistence, Privilege Escalation, or Defense Evasion. Cloud accounts are those created and configured by an organization for use by users, remote support, services, or for administration of resources within a cloud service provider or SaaS application. Cloud Accounts can exist solely in the cloud; alternatively, they may be hybrid-joined between on-premises systems and the cloud through syncing or federation with other identity sources such as Windows Active Directory.

Tactic: Execution
ID: TA0002
Description:
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.

Techniques Under the Execution Tactic

ID	Name	Description
T1651	Cloud Administration Command	Adversaries may abuse cloud management services to execute commands within virtual machines. Resources such as AWS Systems Manager, Azure RunCommand, and Runbooks allow users to remotely run scripts in virtual machines by leveraging installed virtual machine agents.

T1059	Command and Scripting Interpreter	Adversaries may abuse command and script interpreters to execute commands, scripts, or binaries. These interfaces and languages provide ways of interacting with computer systems and are a common feature across many different platforms. Most systems come with some built-in command-line interface and scripting capabilities, for example, macOS and Linux distributions include some flavor of Unix Shell while Windows installations include the Windows Command Shell and PowerShell.
T1059.001	PowerShell	Adversaries may abuse PowerShell commands and scripts for execution. PowerShell is a powerful interactive command-line interface and scripting environment included in the Windows operating system. 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 (though administrator permissions are required to use PowerShell to connect to remote systems).
T1059.002	AppleScript	Adversaries may abuse AppleScript for execution. AppleScript is a macOS scripting language designed to control applications and parts of the OS via inter-application messages called AppleEvents. These AppleEvent messages can be sent independently or easily scripted with AppleScript. These events can locate open windows, send keystrokes, and interact with almost any open application locally or remotely.
T1059.003	Windows Command Shell	Adversaries may abuse the Windows command shell for execution. The Windows command shell (cmd) is the primary command prompt on Windows systems. The Windows command prompt can be used to control almost any aspect of a system, with various permission levels required for different subsets of commands. The command prompt can be invoked remotely via Remote Services such as SSH.

T1059.004	Unix Shell	Adversaries may abuse Unix shell commands and scripts for execution. Unix shells are the primary command prompt on Linux and macOS systems, though many variations of the Unix shell exist (e.g. sh, bash, zsh, etc.) depending on the specific OS or distribution. Unix shells can control every aspect of a system, with certain commands requiring elevated privileges.
T1059.005	Visual Basic	Adversaries may abuse Visual Basic (VB) for execution. VB is a programming language created by Microsoft with interoperability with many Windows technologies such as Component Object Model and the Native API through the Windows API. Although tagged as legacy with no planned future evolutions, VB is integrated and supported in the .NET Framework and cross-platform .NET Core.
T1059.006	Python	Adversaries may abuse Python commands and scripts for execution. Python is a very popular scripting/programming language, with capabilities to perform many functions. Python can be executed interactively from the command-line (via the python.exe interpreter) or via scripts (.py) that can be written and distributed to different systems. Python code can also be compiled into binary executables.
T1059.007	JavaScript	Adversaries may abuse various implementations of JavaScript for execution. JavaScript (JS) is a platform-independent scripting language (compiled just-in-time at runtime) commonly associated with scripts in webpages, though JS can be executed in runtime environments outside the browser.
T1059.008	Network Device CLI	Adversaries may abuse scripting or built-in command line interpreters (CLI) on network devices to execute malicious command and payloads. The CLI is the primary means through which users and administrators interact with the device in order to view system information, modify device operations, or perform diagnostic and administrative functions. CLIs typically contain various permission levels required for different commands.

T1059.009	Cloud API	Adversaries may abuse cloud APIs to execute malicious commands. APIs available in cloud environments provide various functionalities and are a feature-rich method for programmatic access to nearly all aspects of a tenant. These APIs may be utilized through various methods such as command line interpreters (CLIs), in-browser Cloud Shells, PowerShell modules like Azure for PowerShell, or software developer kits (SDKs) available for languages such as Python.
T1059.01	AutoHotKey & AutoIT	Adversaries may execute commands and perform malicious tasks using AutoIT and AutoHotKey automation scripts. AutoIT and AutoHotkey (AHK) are scripting languages that enable users to automate Windows tasks. These automation scripts can be used to perform a wide variety of actions, such as clicking on buttons, entering text, and opening and closing programs.
T1059.011	Lua	Adversaries may abuse Lua commands and scripts for execution. Lua is a cross-platform scripting and programming language primarily designed for embedded use in applications. Lua can be executed on the command-line (through the stand-alone lua interpreter), via scripts (.lua), or from Lua-embedded programs (through the struct lua_State).
T1609	Container Administration Command	Adversaries may abuse a container administration service to execute commands within a container. A container administration service such as the Docker daemon, the Kubernetes API server, or the kubelet may allow remote management of containers within an environment.

T1610	Deploy Container	Adversaries may deploy a container into an environment to facilitate execution or evade defenses. In some cases, adversaries may deploy a new container to execute processes associated with a particular image or deployment, such as processes that execute or download malware. In others, an adversary may deploy a new container configured without network rules, user limitations, etc. to bypass existing defenses within the environment. In Kubernetes environments, an adversary may attempt to deploy a privileged or vulnerable container into a specific node in order to Escape to Host and access other containers running on the node.
T1203	Exploitation for Client Execution	Adversaries may exploit software vulnerabilities in client applications to execute code. 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.
T1559	Inter-Process Communication	Adversaries may abuse inter-process communication (IPC) mechanisms for local code or command execution. IPC is typically used by processes to share data, communicate with each other, or synchronize execution. IPC is also commonly used to avoid situations such as deadlocks, which occurs when processes are stuck in a cyclic waiting pattern.

T1559.001	Component Object Model	Adversaries may use the Windows Component Object Model (COM) for local code execution. COM is an inter-process communication (IPC) component of the native Windows application programming interface (API) that enables interaction between software objects, or executable code that implements one or more interfaces. Through COM, a client object can call methods of server objects, which are typically binary Dynamic Link Libraries (DLL) or executables (EXE). Remote COM execution is facilitated by Remote Services such as Distributed Component Object Model (DCOM).
T1559.002	Dynamic Data Exchange	Adversaries may use Windows Dynamic Data Exchange (DDE) to execute arbitrary commands. 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.
T1559.003	XPC Services	Adversaries can provide malicious content to an XPC service daemon for local code execution. macOS uses XPC services for basic inter-process communication between various processes, such as between the XPC Service daemon and third-party application privileged helper tools. Applications can send messages to the XPC Service daemon, which runs as root, using the low-level XPC Service C API or the high level NSXPCConnection API in order to handle tasks that require elevated privileges (such as network connections). Applications are responsible for providing the protocol definition which serves as a blueprint of the XPC services. Developers typically use XPC Services to provide applications stability and privilege separation between the application client and the daemon.

T1106	Native API	Adversaries may interact with the native OS application programming interface (API) to execute behaviors. Native APIs provide a controlled means of calling low-level OS services within the kernel, such as those involving hardware/devices, memory, and processes. These native APIs are leveraged by the OS during system boot (when other system components are not yet initialized) as well as carrying out tasks and requests during routine operations.
T1053	Scheduled Task/Job	Adversaries may abuse task scheduling functionality to facilitate initial or recurring execution of malicious code. Utilities exist within all major operating systems to schedule programs or scripts to be executed at a specified date and time. A task can also be scheduled on a remote system, provided the proper authentication is met (ex: RPC and file and printer sharing in Windows environments). Scheduling a task on a remote system typically may require being a member of an admin or otherwise privileged group on the remote system.
T1053.002	At	Adversaries may abuse the at utility to perform task scheduling for initial or recurring execution of malicious code. The at utility exists as an executable within Windows, Linux, and macOS for scheduling tasks at a specified time and date. Although deprecated in favor of Scheduled Task's schtasks in Windows environments, using at requires that the Task Scheduler service be running, and the user to be logged on as a member of the local Administrators group. In addition to explicitly running the at command, adversaries may also schedule a task with at by directly leveraging the Windows Management Instrumentation Win32_ScheduledJob WMI class.
T1053.003	Cron	Adversaries may abuse the cron utility to perform task scheduling for initial or recurring execution of malicious code. The cron utility is a time-based job scheduler for Unix-like operating systems. The crontab file contains the schedule of cron entries to be run and the specified times for execution. Any crontab files are stored in operating system-specific file paths.

T1053.005	Scheduled Task	Adversaries may abuse the Windows Task Scheduler to perform task scheduling for initial or recurring execution of malicious code. There are multiple ways to access the Task Scheduler in Windows. The schtasks utility can be run directly on the command line, or the Task Scheduler can be opened through the GUI within the Administrator Tools section of the Control Panel. In some cases, adversaries have used a .NET wrapper for the Windows Task Scheduler, and alternatively, adversaries have used the Windows netapi32 library and Windows Management Instrumentation (WMI) to create a scheduled task. Adversaries may also utilize the Powershell Cmdlet Invoke-CimMethod, which leverages WMI class PS_ScheduledTask to create a scheduled task via an XML path.
T1053.006	Systemd Timers	Adversaries may abuse systemd timers to perform task scheduling for initial or recurring execution of malicious code. Systemd timers are unit files with file extension .timer that control services. Timers can be set to run on a calendar event or after a time span relative to a starting point. They can be used as an alternative to Cron in Linux environments. Systemd timers may be activated remotely via the systemctl command line utility, which operates over SSH.
T1053.007	Container Orchestration Job	Adversaries may abuse task scheduling functionality provided by container orchestration tools such as Kubernetes to schedule deployment of containers configured to execute malicious code. Container orchestration jobs run these automated tasks at a specific date and time, similar to cron jobs on a Linux system. Deployments of this type can also be configured to maintain a quantity of containers over time, automating the process of maintaining persistence within a cluster.
T1648	Serverless Execution	Adversaries may abuse serverless computing, integration, and automation services to execute arbitrary code in cloud environments. Many cloud providers offer a variety of serverless resources, including compute engines, application integration services, and web servers.

T1129	Shared Modules	Adversaries may execute malicious payloads via loading shared modules. Shared modules are executable files that are loaded into processes to provide access to reusable code, such as specific custom functions or invoking OS API functions (i.e., Native API).
T1072	Software Deployment Tools	Adversaries may gain access to and use centralized software suites installed within an enterprise to execute commands and move laterally through the network. Configuration management and software deployment applications may be used in an enterprise network or cloud environment for routine administration purposes. These systems may also be integrated into CI/CD pipelines. Examples of such solutions include: SCCM, HBSS, Altiris, AWS Systems Manager, Microsoft Intune, Azure Arc, and GCP Deployment Manager.
T1569	System Services	Adversaries may abuse system services or daemons to execute commands or programs. Adversaries can execute malicious content by interacting with or creating services either locally or remotely. Many services are set to run at boot, which can aid in achieving persistence (Create or Modify System Process), but adversaries can also abuse services for one-time or temporary execution.
T1569.001	Launchctl	Adversaries may abuse launchctl to execute commands or programs. Launchctl interfaces with launchd, the service management framework for macOS. Launchctl supports taking subcommands on the command-line, interactively, or even redirected from standard input.
T1569.002	Service Execution	Adversaries may abuse the Windows service control manager to execute malicious commands or payloads. The Windows service control manager (services.exe) is an interface to manage and manipulate services. The service control manager is accessible to users via GUI components as well as system utilities such as sc.exe and Net.

T1204	User Execution	An adversary may rely upon specific actions by a user in order to gain execution. Users may be subjected to social engineering to get them to execute malicious code by, for example, opening a malicious document file or link. These user actions will typically be observed as follow-on behavior from forms of Phishing.
T1204.001	Malicious Link	An adversary may rely upon a user clicking a malicious link in order to gain execution. Users may be subjected to social engineering to get them to click on a link that will lead to code execution. This user action will typically be observed as follow-on behavior from Spearphishing Link. Clicking on a link may also lead to other execution techniques such as exploitation of a browser or application vulnerability via Exploitation for Client Execution. Links may also lead users to download files that require execution via Malicious File.
T1204.002	Malicious File	An adversary may rely upon a user opening a malicious file in order to gain execution. Users may be subjected to social engineering to get them to open a file that will lead to code execution. This user action will typically be observed as follow-on behavior from Spearphishing Attachment. Adversaries may use several types of files that require a user to execute them, including .doc, .pdf, .xls, .rtf, .scr, .exe, .lnk, .pif, .cpl, and .reg.
T1204.003	Malicious Image	Adversaries may rely on a user running a malicious image to facilitate execution. Amazon Web Services (AWS) Amazon Machine Images (AMIs), Google Cloud Platform (GCP) Images, and Azure Images as well as popular container runtimes such as Docker can be backdoored. Backdoored images may be uploaded to a public repository via Upload Malware, and users may then download and deploy an instance or container from the image without realizing the image is malicious, thus bypassing techniques that specifically achieve Initial Access. This can lead to the execution of malicious code, such as code that executes cryptocurrency mining, in the instance or container.

T1047	Windows Management Instrumentation	Adversaries may abuse Windows Management Instrumentation (WMI) to execute malicious commands and payloads. WMI is designed for programmers and is the infrastructure for management data and operations on Windows systems. WMI is an administration feature that provides a uniform environment to access Windows system components.

Tactic: Persistence
ID: TA0003
Description:
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.

Techniques Under the Initial Access Tactic

ID	Name	Description
T1098	Account Manipulation	Adversaries may manipulate accounts to maintain and/or elevate access to victim systems. Account manipulation may consist of any action that preserves or modifies adversary access to a compromised account, such as modifying credentials or permission groups. These actions could also include account activity designed to subvert security policies, such as performing iterative password updates to bypass password duration policies and preserve the life of compromised credentials.
T1098.001	Additional Cloud Credentials	Adversaries may add adversary-controlled credentials to a cloud account to maintain persistent access to victim accounts and instances within the environment.

T1098.002	Additional Email Delegate Permissions	Adversaries may grant additional permission levels to maintain persistent access to an adversary-controlled email account.
T1098.003	Additional Cloud Roles	An adversary may add additional roles or permissions to an adversary-controlled cloud account to maintain persistent access to a tenant. For example, adversaries may update IAM policies in cloud-based environments or add a new global administrator in Office 365 environments. With sufficient permissions, a compromised account can gain almost unlimited access to data and settings (including the ability to reset the passwords of other admins).
T1098.004	SSH Authorized Keys	Adversaries may modify the SSH authorized_keys file to maintain persistence on a victim host. Linux distributions and macOS commonly use key-based authentication to secure the authentication process of SSH sessions for remote management. The authorized_keys file in SSH specifies the SSH keys that can be used for logging into the user account for which the file is configured. This file is usually found in the user's home directory under <user-home>/.ssh/authorized_keys. Users may edit the system’s SSH config file to modify the directives PubkeyAuthentication and RSAAuthentication to the value "yes" to ensure public key and RSA authentication are enabled. The SSH config file is usually located under /etc/ssh/sshd_config.
T1098.005	Device Registration	Adversaries may register a device to an adversary-controlled account. Devices may be registered in a multifactor authentication (MFA) system, which handles authentication to the network, or in a device management system, which handles device access and compliance.

T1098.006	Additional Container Cluster Roles	An adversary may add additional roles or permissions to an adversary-controlled user or service account to maintain persistent access to a container orchestration system. For example, an adversary with sufficient permissions may create a RoleBinding or a ClusterRoleBinding to bind a Role or ClusterRole to a Kubernetes account. Where attribute-based access control (ABAC) is in use, an adversary with sufficient permissions may modify a Kubernetes ABAC policy to give the target account additional permissions.
T1098.007	Additional Local or Domain Groups	An adversary may add additional local or domain groups to an adversary-controlled account to maintain persistent access to a system or domain.
T1197	BITS Jobs	Adversaries may abuse BITS jobs to persistently execute code and perform various background tasks. Windows Background Intelligent Transfer Service (BITS) is a low-bandwidth, asynchronous file transfer mechanism exposed through Component Object Model (COM). 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.
T1547	Boot or Logon Autostart Execution	Adversaries may configure system settings to automatically execute a program during system boot or logon to maintain persistence or gain higher-level privileges on compromised systems. Operating systems may have mechanisms for automatically running a program on system boot or account logon. These mechanisms may include automatically executing programs that are placed in specially designated directories or are referenced by repositories that store configuration information, such as the Windows Registry. An adversary may achieve the same goal by modifying or extending features of the kernel.

T1547.001	Registry Run Keys / Startup Folder	Adversaries may achieve persistence by adding a program to a startup folder or referencing it with a Registry run key. Adding an entry to the "run keys" in the Registry or startup folder will cause the program referenced to be executed when a user logs in. These programs will be executed under the context of the user and will have the account's associated permissions level.
T1547.002	Authentication Package	Adversaries may abuse authentication packages to execute DLLs when the system boots. 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.
T1547.003	Time Providers	Adversaries may abuse time providers to execute DLLs when the system boots. The Windows Time service (W32Time) enables time synchronization across and within domains. W32Time time providers are responsible for retrieving time stamps from hardware/network resources and outputting these values to other network clients.
T1547.004	Winlogon Helper DLL	Adversaries may abuse features of Winlogon to execute DLLs and/or executables when a user logs in. 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 HKLM\Software[\Wow6432Node\]\Microsoft\Windows NT\CurrentVersion\Winlogon\ and HKCU\Software\Microsoft\Windows NT\CurrentVersion\Winlogon\ are used to manage additional helper programs and functionalities that support Winlogon.
T1547.005	Security Support Provider	Adversaries may abuse security support providers (SSPs) to execute DLLs when the system boots. Windows 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.

T1547.006	Kernel Modules and Extensions	Adversaries may modify the kernel to automatically execute programs on system boot. Loadable Kernel Modules (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.
T1547.007	Re-opened Applications	Adversaries may modify plist files to automatically run an application when a user logs in. When a user logs out or restarts via the macOS Graphical User Interface (GUI), a prompt is provided to the user with a checkbox to "Reopen windows when logging back in". When selected, all applications currently open are added to a property list file named com.apple.loginwindow.[UUID].plist within the ~/Library/Preferences/ByHost directory. Applications listed in this file are automatically reopened upon the user’s next logon.
T1547.008	LSASS Driver	Adversaries may modify or add LSASS drivers to obtain persistence on compromised systems. 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.
T1547.009	Shortcut Modification	Adversaries may create or modify shortcuts that can execute a program during system boot or user login. Shortcuts or symbolic links are used to reference other files or programs that will be opened or executed when the shortcut is clicked or executed by a system startup process.

T1547.01	Port Monitors	Adversaries may use port monitors to run an adversary supplied DLL during system boot for persistence or privilege escalation. A port monitor can be set through the AddMonitor API call to set a DLL to be loaded at startup. This DLL can be located in C:\Windows\System32 and will be loaded and run by the print spooler service, spoolsv.exe, under SYSTEM level permissions on boot.
T1547.012	Print Processors	Adversaries may abuse print processors to run malicious DLLs during system boot for persistence and/or privilege escalation. Print processors are DLLs that are loaded by the print spooler service, spoolsv.exe, during boot.
T1547.013	XDG Autostart Entries	Adversaries may add or modify XDG Autostart Entries to execute malicious programs or commands when a user's desktop environment is loaded at login. XDG Autostart entries are available for any XDG-compliant Linux system. XDG Autostart entries use Desktop Entry files (.desktop) to configure the user's desktop environment upon user login. These configuration files determine what applications launch upon user login, define associated applications to open specific file types, and define applications used to open removable media.
T1547.014	Active Setup	Adversaries may achieve persistence by adding a Registry key to the Active Setup of the local machine. Active Setup is a Windows mechanism that is used to execute programs when a user logs in. The value stored in the Registry key will be executed after a user logs into the computer. These programs will be executed under the context of the user and will have the account's associated permissions level.

T1547.015	Login Items	Adversaries may add login items to execute upon user login to gain persistence or escalate privileges. Login items are applications, documents, folders, or server connections that are automatically launched when a user logs in. Login items can be added via a shared file list or Service Management Framework. Shared file list login items can be set using scripting languages such as AppleScript, whereas the Service Management Framework uses the API call SMLoginItemSetEnabled.
T1037	Boot or Logon Initialization Scripts	Adversaries may use scripts automatically executed at boot or logon initialization to establish persistence. Initialization scripts can be used to perform administrative functions, which may often execute other programs or send information to an internal logging server. These scripts can vary based on operating system and whether applied locally or remotely.
T1037.001	Logon Script (Windows)	Adversaries may use Windows logon scripts automatically executed at logon initialization to establish persistence. Windows allows logon scripts to be run whenever a specific user or group of users log into a system. This is done via adding a path to a script to the HKCU\Environment\UserInitMprLogonScript Registry key.
T1037.002	Login Hook	Adversaries may use a Login Hook to establish persistence executed upon user logon. A login hook is a plist file that points to a specific script to execute with root privileges upon user logon. The plist file is located in the /Library/Preferences/com.apple.loginwindow.plist file and can be modified using the defaults command-line utility. This behavior is the same for logout hooks where a script can be executed upon user logout. All hooks require administrator permissions to modify or create hooks.

T1037.003	Network Logon Script	Adversaries may use network logon scripts automatically executed at logon initialization to establish persistence. Network logon scripts can be assigned using Active Directory or Group Policy Objects. These logon scripts run with the privileges of the user they are assigned to. Depending on the systems within the network, initializing one of these scripts could apply to more than one or potentially all systems.
T1037.004	RC Scripts	Adversaries may establish persistence by modifying RC scripts which are executed during a Unix-like system’s startup. These files allow system administrators to map and start custom services at startup for different run levels. RC scripts require root privileges to modify.
T1037.005	Startup Items	Adversaries may use startup items automatically executed at boot initialization to establish persistence. 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.
T1176	Browser Extensions	Adversaries may abuse Internet browser extensions to establish persistent access to victim systems. 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 and generally have access and permissions to everything that the browser can access.
T1554	Compromise Host Software Binary	Adversaries may modify host software binaries to establish persistent access to systems. Software binaries/executables provide a wide range of system commands or services, programs, and libraries. Common software binaries are SSH clients, FTP clients, email clients, web browsers, and many other user or server applications.
T1136	Create Account	Adversaries may create an account to maintain access to victim systems. With a sufficient level of access, creating such accounts may be used to establish secondary credentialed access that do not require persistent remote access tools to be deployed on the system.

T1136.001	Local Account	Adversaries may create a local account to maintain access to victim systems. Local accounts are those configured by an organization for use by users, remote support, services, or for administration on a single system or service.
T1136.002	Domain Account	Adversaries may create a domain account to maintain access to victim systems. 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 user, administrator, and service accounts. With a sufficient level of access, the net user /add /domain command can be used to create a domain account.
T1136.003	Cloud Account	Adversaries may create a cloud account to maintain access to victim systems. With a sufficient level of access, such accounts may be used to establish secondary credentialed access that does not require persistent remote access tools to be deployed on the system.
T1543	Create or Modify System Process	Adversaries may create or modify system-level processes to repeatedly execute malicious payloads as part of persistence. When operating systems boot up, they can start processes that perform background system functions. On Windows and Linux, these system processes are referred to as services. On macOS, launchd processes known as Launch Daemon and Launch Agent are run to finish system initialization and load user specific parameters.

T1543.001	Launch Agent	Adversaries may create or modify launch agents to repeatedly execute malicious payloads as part of persistence. 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) file found in /System/Library/LaunchAgents, /Library/LaunchAgents, and ~/Library/LaunchAgents. Property list files use the Label, ProgramArguments , and RunAtLoad keys to identify the Launch Agent's name, executable location, and execution time. Launch Agents are often installed to perform updates to programs, launch user specified programs at login, or to conduct other developer tasks.
T1543.002	Systemd Service	Adversaries may create or modify systemd services to repeatedly execute malicious payloads as part of persistence. Systemd is a system and service manager commonly used for managing background daemon processes (also known as services) and other system resources. Systemd is the default initialization (init) system on many Linux distributions replacing legacy init systems, including SysVinit and Upstart, while remaining backwards compatible.
T1543.003	Windows Service	Adversaries may create or modify Windows services to repeatedly execute malicious payloads as part of persistence. When Windows boots up, it starts programs or applications called services that perform background system functions. Windows service configuration information, including the file path to the service's executable or recovery programs/commands, is stored in the Windows Registry.

T1543.004	Launch Daemon	Adversaries may create or modify Launch Daemons to execute malicious payloads as part of persistence. Launch Daemons are plist files used to interact with Launchd, the service management framework used by macOS. Launch Daemons require elevated privileges to install, are executed for every user on a system prior to login, and run in the background without the need for user interaction. During the macOS initialization startup, the launchd process loads the parameters for launch-on-demand system-level daemons from plist files found in /System/Library/LaunchDaemons/ and /Library/LaunchDaemons/. Required Launch Daemons parameters include a Label to identify the task, Program to provide a path to the executable, and RunAtLoad to specify when the task is run. Launch Daemons are often used to provide access to shared resources, updates to software, or conduct automation tasks.
T1543.005	Container Service	Adversaries may create or modify container or container cluster management tools that run as daemons, agents, or services on individual hosts. These include software for creating and managing individual containers, such as Docker and Podman, as well as container cluster node-level agents such as kubelet. By modifying these services, an adversary may be able to achieve persistence or escalate their privileges on a host.
T1546	Event Triggered Execution	Adversaries may establish persistence and/or elevate privileges using system mechanisms that trigger execution based on specific events. Various operating systems have means to monitor and subscribe to events such as logons or other user activity such as running specific applications/binaries. Cloud environments may also support various functions and services that monitor and can be invoked in response to specific cloud events.

T1546.001	Change Default File Association	Adversaries may establish persistence by executing malicious content triggered by a file type 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 or by administrators using the built-in assoc utility. 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.
T1546.002	Screensaver	Adversaries may establish persistence by executing malicious content triggered by user inactivity. Screensavers are programs that execute after a configurable time of user inactivity and consist of Portable Executable (PE) files with a .scr file extension. The Windows screensaver application scrnsave.scr is located in C:\Windows\System32\, and C:\Windows\sysWOW64\ on 64-bit Windows systems, along with screensavers included with base Windows installations.
T1546.003	Windows Management Instrumentation Event Subscription	Adversaries may establish persistence and elevate privileges by executing malicious content triggered by a Windows Management Instrumentation (WMI) event subscription. WMI can be used to install event filters, providers, consumers, and bindings that execute code when a defined event occurs. Examples of events that may be subscribed to are the wall clock time, user login, or the computer's uptime.

T1546.004	Unix Shell Configuration Modification	Adversaries may establish persistence through executing malicious commands triggered by a user's shell. User Unix Shells execute several configuration scripts at different points throughout the session based on events. For example, when a user opens a command-line interface or remotely logs in (such as via SSH) a login shell is initiated. The login shell executes scripts from the system (/etc) and the user's home directory (~/) to configure the environment. All login shells on a system use /etc/profile when initiated. These configuration scripts run at the permission level of their directory and are often used to set environment variables, create aliases, and customize the user's environment. When the shell exits or terminates, additional shell scripts are executed to ensure the shell exits appropriately.
T1546.005	Trap	Adversaries may establish persistence by executing malicious content triggered by an interrupt signal. The trap 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 ctrl+c and ctrl+d.
T1546.006	LC_LOAD_DYLIB Addition	Adversaries may establish persistence by executing malicious content triggered by the execution of tainted binaries. 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 as adjustments are made to the rest of the fields and dependencies. There are tools available to perform these changes.

T1546.007	Netsh Helper DLL	Adversaries may establish persistence by executing malicious content triggered by Netsh Helper DLLs. 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. The paths to registered netsh.exe helper DLLs are entered into the Windows Registry at HKLM\SOFTWARE\Microsoft\Netsh.
T1546.008	Accessibility Features	Adversaries may establish persistence and/or elevate privileges by executing malicious content triggered by accessibility features. Windows contains accessibility features that may be launched with a key combination before a user has logged in (ex: 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.
T1546.009	AppCert DLLs	Adversaries may establish persistence and/or elevate privileges by executing malicious content triggered by AppCert DLLs loaded into processes. Dynamic-link libraries (DLLs) that are specified in the AppCertDLLs Registry key under HKEY_LOCAL_MACHINE\System\CurrentControlSet\Control\Session Manager\ are loaded into every process that calls the ubiquitously used application programming interface (API) functions CreateProcess, CreateProcessAsUser, CreateProcessWithLoginW, CreateProcessWithTokenW, or WinExec.

T1546.01	AppInit DLLs	Adversaries may establish persistence and/or elevate privileges by executing malicious content triggered by AppInit DLLs loaded into processes. Dynamic-link libraries (DLLs) that are specified in the AppInit_DLLs value in the Registry keys HKEY_LOCAL_MACHINE\Software\Microsoft\Windows NT\CurrentVersion\Windows or HKEY_LOCAL_MACHINE\Software\Wow6432Node\Microsoft\Windows NT\CurrentVersion\Windows 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.
T1546.011	Application Shimming	Adversaries may establish persistence and/or elevate privileges by executing malicious content triggered by application shims. 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.
T1546.012	Image File Execution Options Injection	Adversaries may establish persistence and/or elevate privileges by executing malicious content triggered by Image File Execution Options (IFEO) debuggers. IFEOs 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).
T1546.013	PowerShell Profile	Adversaries may gain persistence and elevate privileges by executing malicious content triggered by PowerShell profiles. A PowerShell profile (profile.ps1) is a script that runs when PowerShell starts and can be used as a logon script to customize user environments.

T1546.014	Emond	Adversaries may gain persistence and elevate privileges by executing malicious content triggered by the Event Monitor Daemon (emond). Emond is a Launch Daemon that accepts events from various services, runs them through a simple rules engine, and takes action. The emond binary at /sbin/emond will load any rules from the /etc/emond.d/rules/ directory and take action once an explicitly defined event takes place.
T1546.015	Component Object Model Hijacking	Adversaries may establish persistence by executing malicious content triggered by hijacked references to Component Object Model (COM) objects. COM is a system within Windows to enable interaction between software components through the operating system. References to various COM objects are stored in the Registry.
T1546.016	Installer Packages	Adversaries may establish persistence and elevate privileges by using an installer to trigger the execution of malicious content. Installer packages are OS specific and contain the resources an operating system needs to install applications on a system. Installer packages can include scripts that run prior to installation as well as after installation is complete. Installer scripts may inherit elevated permissions when executed. Developers often use these scripts to prepare the environment for installation, check requirements, download dependencies, and remove files after installation.

T1546.017	Udev Rules	Adversaries may maintain persistence through executing malicious content triggered using udev rules. Udev is the Linux kernel device manager that dynamically manages device nodes, handles access to pseudo-device files in the /dev directory, and responds to hardware events, such as when external devices like hard drives or keyboards are plugged in or removed. Udev uses rule files with match keys to specify the conditions a hardware event must meet and action keys to define the actions that should follow. Root permissions are required to create, modify, or delete rule files located in /etc/udev/rules.d/, /run/udev/rules.d/, /usr/lib/udev/rules.d/, /usr/local/lib/udev/rules.d/, and /lib/udev/rules.d/. Rule priority is determined by both directory and by the digit prefix in the rule filename.
T1133	External Remote Services	Adversaries may leverage external-facing remote services to initially access and/or persist within a network. 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 and VNC can also be used externally.
T1574	Hijack Execution Flow	Adversaries may execute their own malicious payloads by hijacking the way operating systems run programs. Hijacking execution flow can be for the purposes of persistence, since this hijacked execution may reoccur over time. Adversaries may also use these mechanisms to elevate privileges or evade defenses, such as application control or other restrictions on execution.
T1574.001	DLL Search Order Hijacking	Adversaries may execute their own malicious payloads by hijacking the search order used to load DLLs. Windows systems use a common method to look for required DLLs to load into a program. Hijacking DLL loads may be for the purpose of establishing persistence as well as elevating privileges and/or evading restrictions on file execution.

T1574.002	DLL Side-Loading	Adversaries may execute their own malicious payloads by side-loading DLLs. Similar to DLL Search Order Hijacking, side-loading involves hijacking which DLL a program loads. But rather than just planting the DLL within the search order of a program then waiting for the victim application to be invoked, adversaries may directly side-load their payloads by planting then invoking a legitimate application that executes their payload(s).
T1574.004	Dylib Hijacking	Adversaries may execute their own payloads by placing a malicious dynamic library (dylib) with an expected name in a path a victim application searches at runtime. The dynamic loader will try to find the dylibs based on the sequential order of the search paths. Paths to dylibs may be prefixed with @rpath, which allows developers to use relative paths to specify an array of search paths used at runtime based on the location of the executable. Additionally, if weak linking is used, such as the LC_LOAD_WEAK_DYLIB function, an application will still execute even if an expected dylib is not present. Weak linking enables developers to run an application on multiple macOS versions as new APIs are added.
T1574.005	Executable Installer File Permissions Weakness	Adversaries may execute their own malicious payloads by hijacking the binaries used by an installer. These 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.

T1574.006	Dynamic Linker Hijacking	Adversaries may execute their own malicious payloads by hijacking environment variables the dynamic linker uses to load shared libraries. During the execution preparation phase of a program, the dynamic linker loads specified absolute paths of shared libraries from environment variables and files, such as LD_PRELOAD on Linux or DYLD_INSERT_LIBRARIES on macOS. Libraries specified in environment variables are loaded first, taking precedence over system libraries with the same function name. These variables are often used by developers to debug binaries without needing to recompile, deconflict mapped symbols, and implement custom functions without changing the original library.
T1574.007	Path Interception by PATH Environment Variable	Adversaries may execute their own malicious payloads by hijacking environment variables used to load libraries. The PATH environment variable contains a list of directories (User and System) that the OS searches sequentially through in search of the binary that was called from a script or the command line.
T1574.008	Path Interception by Search Order Hijacking	Adversaries may execute their own malicious payloads by hijacking the search order used to load other programs. Because some programs do not call other programs using the full path, adversaries may place their own file in the directory where the calling program is located, causing the operating system to launch their malicious software at the request of the calling program.
T1574.009	Path Interception by Unquoted Path	Adversaries may execute their own malicious payloads by hijacking vulnerable file path references. Adversaries can take advantage of paths that lack surrounding quotations by placing an executable in a higher level directory within the path, so that Windows will choose the adversary's executable to launch.

T1574.01	Services File Permissions Weakness	Adversaries may execute their own malicious payloads by hijacking the binaries used by services. Adversaries may use flaws in the permissions of Windows services to replace the binary that is executed upon service start. These service 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.
T1574.011	Services Registry Permissions Weakness	Adversaries may execute their own malicious payloads by hijacking the Registry entries used by services. Adversaries may use flaws in the permissions for Registry keys related to services to redirect from the originally specified executable to one that they control, in order to launch their own code when a service starts. Windows stores local service configuration information in the Registry under HKLM\SYSTEM\CurrentControlSet\Services. 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, or Reg. Access to Registry keys is controlled through access control lists and user permissions.
T1574.012	COR_PROFILER	Adversaries may leverage the COR_PROFILER environment variable to hijack the execution flow of programs that load the .NET CLR. The COR_PROFILER is a .NET Framework feature which allows developers to specify an unmanaged (or external of .NET) profiling DLL to be loaded into each .NET process that loads the Common Language Runtime (CLR). These profilers are designed to monitor, troubleshoot, and debug managed code executed by the .NET CLR.

T1574.013	KernelCallbackTable	Adversaries may abuse the KernelCallbackTable of a process to hijack its execution flow in order to run their own payloads. The KernelCallbackTable can be found in the Process Environment Block (PEB) and is initialized to an array of graphic functions available to a GUI process once user32.dll is loaded.
T1574.014	AppDomainManager	Adversaries may execute their own malicious payloads by hijacking how the .NET AppDomainManager loads assemblies. The .NET framework uses the AppDomainManager class to create and manage one or more isolated runtime environments (called application domains) inside a process to host the execution of .NET applications. Assemblies (.exe or .dll binaries compiled to run as .NET code) may be loaded into an application domain as executable code.
T1525	Implant Internal Image	Adversaries may implant cloud or container images with malicious code to establish persistence after gaining access to an environment. Amazon Web Services (AWS) Amazon Machine Images (AMIs), Google Cloud Platform (GCP) Images, and Azure Images as well as popular container runtimes such as Docker can be implanted or backdoored. Unlike Upload Malware, this technique focuses on adversaries implanting an image in a registry within a victim's environment. 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.

T1556	Modify Authentication Process	Adversaries may modify authentication mechanisms and processes to access user credentials or enable otherwise unwarranted access to accounts. The authentication process is handled by mechanisms, such as the Local Security Authentication Server (LSASS) process and the Security Accounts Manager (SAM) on Windows, pluggable authentication modules (PAM) on Unix-based systems, and authorization plugins on MacOS systems, responsible for gathering, storing, and validating credentials. By modifying an authentication process, an adversary may be able to authenticate to a service or system without using Valid Accounts.
T1556.001	Domain Controller Authentication	Adversaries may patch the authentication process on a domain controller to bypass the typical authentication mechanisms and enable access to accounts.
T1556.002	Password Filter DLL	Adversaries may register malicious password filter dynamic link libraries (DLLs) into the authentication process to acquire user credentials as they are validated.
T1556.003	Pluggable Authentication Modules	Adversaries may modify pluggable authentication modules (PAM) to access user credentials or enable otherwise unwarranted access to accounts. PAM is a modular system of configuration files, libraries, and executable files which guide authentication for many services. The most common authentication module is pam_unix.so, which retrieves, sets, and verifies account authentication information in /etc/passwd and /etc/shadow.
T1556.004	Network Device Authentication	Adversaries may use Patch System Image to hard code a password in the operating system, thus bypassing of native authentication mechanisms for local accounts on network devices.

T1556.005	Reversible Encryption	An adversary may abuse Active Directory authentication encryption properties to gain access to credentials on Windows systems. The AllowReversiblePasswordEncryption property specifies whether reversible password encryption for an account is enabled or disabled. By default this property is disabled (instead storing user credentials as the output of one-way hashing functions) and should not be enabled unless legacy or other software require it.
T1556.006	Multi-Factor Authentication	Adversaries may disable or modify multi-factor authentication (MFA) mechanisms to enable persistent access to compromised accounts.
T1556.007	Hybrid Identity	Adversaries may patch, modify, or otherwise backdoor cloud authentication processes that are tied to on-premises user identities in order to bypass typical authentication mechanisms, access credentials, and enable persistent access to accounts.
T1556.008	Network Provider DLL	Adversaries may register malicious network provider dynamic link libraries (DLLs) to capture cleartext user credentials during the authentication process. Network provider DLLs allow Windows to interface with specific network protocols and can also support add-on credential management functions. During the logon process, Winlogon (the interactive logon module) sends credentials to the local mpnotify.exe process via RPC. The mpnotify.exe process then shares the credentials in cleartext with registered credential managers when notifying that a logon event is happening.
T1556.009	Conditional Access Policies	Adversaries may disable or modify conditional access policies to enable persistent access to compromised accounts. Conditional access policies are additional verifications used by identity providers and identity and access management systems to determine whether a user should be granted access to a resource.

T1137	Office Application Startup	Adversaries may leverage Microsoft Office-based applications for persistence between startups. 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; this can include the use of Office Template Macros and add-ins.
T1137.001	Office Template Macros	Adversaries may abuse Microsoft Office templates to obtain persistence on a compromised system. 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.
T1137.002	Office Test	Adversaries may abuse the Microsoft Office "Office Test" Registry key to obtain persistence on a compromised system. An Office Test Registry location exists that allows a user to specify an arbitrary DLL that will be executed every time an Office application is started. This Registry key is thought to be used by Microsoft to load DLLs for testing and debugging purposes while developing Office applications. This Registry key is not created by default during an Office installation.
T1137.003	Outlook Forms	Adversaries may abuse Microsoft Outlook forms to obtain persistence on a compromised system. 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.
T1137.004	Outlook Home Page	Adversaries may abuse Microsoft Outlook's Home Page feature to obtain persistence on a compromised system. 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.

T1137.005	Outlook Rules	Adversaries may abuse Microsoft Outlook rules to obtain persistence on a compromised system. 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.
T1137.006	Add-ins	Adversaries may abuse Microsoft Office add-ins to obtain persistence on a compromised system. Office add-ins can be used to add functionality to Office programs. 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.
T1653	Power Settings	Adversaries may impair a system's ability to hibernate, reboot, or shut down in order to extend access to infected machines. When a computer enters a dormant state, some or all software and hardware may cease to operate which can disrupt malicious activity.
T1542	Pre-OS Boot	Adversaries may abuse Pre-OS Boot mechanisms as a way to establish persistence on a system. During the booting process of a computer, firmware and various startup services are loaded before the operating system. These programs control flow of execution before the operating system takes control.
T1542.001	System Firmware	Adversaries may modify system firmware to persist on systems.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.

T1542.002	Component Firmware	Adversaries may modify component firmware to persist on systems. 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 but conducted upon other system components/devices that may not have the same capability or level of integrity checking.
T1542.003	Bootkit	Adversaries may use bootkits to persist on systems. Bootkits reside at a layer below the operating system and may make it difficult to perform full remediation unless an organization suspects one was used and can act accordingly.
T1542.004	ROMMONkit	Adversaries may abuse the ROM Monitor (ROMMON) by loading an unauthorized firmware with adversary code to provide persistent access and manipulate device behavior that is difficult to detect.
T1542.005	TFTP Boot	Adversaries may abuse netbooting to load an unauthorized network device operating system from a Trivial File Transfer Protocol (TFTP) server. TFTP boot (netbooting) is commonly used by network administrators to load configuration-controlled network device images from a centralized management server. Netbooting is one option in the boot sequence and can be used to centralize, manage, and control device images.
T1053	Scheduled Task/Job	Adversaries may abuse task scheduling functionality to facilitate initial or recurring execution of malicious code. Utilities exist within all major operating systems to schedule programs or scripts to be executed at a specified date and time. A task can also be scheduled on a remote system, provided the proper authentication is met (ex: RPC and file and printer sharing in Windows environments). Scheduling a task on a remote system typically may require being a member of an admin or otherwise privileged group on the remote system.

T1053.002	At	Adversaries may abuse the at utility to perform task scheduling for initial or recurring execution of malicious code. The at utility exists as an executable within Windows, Linux, and macOS for scheduling tasks at a specified time and date. Although deprecated in favor of Scheduled Task's schtasks in Windows environments, using at requires that the Task Scheduler service be running, and the user to be logged on as a member of the local Administrators group. In addition to explicitly running the at command, adversaries may also schedule a task with at by directly leveraging the Windows Management Instrumentation Win32_ScheduledJob WMI class.
T1053.003	Cron	Adversaries may abuse the cron utility to perform task scheduling for initial or recurring execution of malicious code. The cron utility is a time-based job scheduler for Unix-like operating systems. The crontab file contains the schedule of cron entries to be run and the specified times for execution. Any crontab files are stored in operating system-specific file paths.
T1053.005	Scheduled Task	Adversaries may abuse the Windows Task Scheduler to perform task scheduling for initial or recurring execution of malicious code. There are multiple ways to access the Task Scheduler in Windows. The schtasks utility can be run directly on the command line, or the Task Scheduler can be opened through the GUI within the Administrator Tools section of the Control Panel. In some cases, adversaries have used a .NET wrapper for the Windows Task Scheduler, and alternatively, adversaries have used the Windows netapi32 library and Windows Management Instrumentation (WMI) to create a scheduled task. Adversaries may also utilize the Powershell Cmdlet Invoke-CimMethod, which leverages WMI class PS_ScheduledTask to create a scheduled task via an XML path.

T1053.006	Systemd Timers	Adversaries may abuse systemd timers to perform task scheduling for initial or recurring execution of malicious code. Systemd timers are unit files with file extension .timer that control services. Timers can be set to run on a calendar event or after a time span relative to a starting point. They can be used as an alternative to Cron in Linux environments. Systemd timers may be activated remotely via the systemctl command line utility, which operates over SSH.
T1053.007	Container Orchestration Job	Adversaries may abuse task scheduling functionality provided by container orchestration tools such as Kubernetes to schedule deployment of containers configured to execute malicious code. Container orchestration jobs run these automated tasks at a specific date and time, similar to cron jobs on a Linux system. Deployments of this type can also be configured to maintain a quantity of containers over time, automating the process of maintaining persistence within a cluster.
T1505	Server Software Component	Adversaries may abuse legitimate extensible development features of servers to establish persistent access to systems. Enterprise server applications may include features that allow developers to write and install software or scripts to extend the functionality of the main application. Adversaries may install malicious components to extend and abuse server applications.
T1505.001	SQL Stored Procedures	Adversaries may abuse SQL stored procedures to establish persistent access to systems. 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).

T1505.002	Transport Agent	Adversaries may abuse Microsoft transport agents to establish persistent access to systems. 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. 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.
T1505.003	Web Shell	Adversaries may backdoor web servers with web shells to establish persistent access to systems. A Web shell is a Web script that is placed on an openly accessible Web server to allow an adversary to access 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.
T1505.004	IIS Components	Adversaries may install malicious components that run on Internet Information Services (IIS) web servers to establish persistence. IIS provides several mechanisms to extend the functionality of the web servers. For example, Internet Server Application Programming Interface (ISAPI) extensions and filters can be installed to examine and/or modify incoming and outgoing IIS web requests. Extensions and filters are deployed as DLL files that export three functions: Get{Extension/Filter}Version, Http{Extension/Filter}Proc, and (optionally) Terminate{Extension/Filter}. IIS modules may also be installed to extend IIS web servers.

T1505.005	Terminal Services DLL	Adversaries may abuse components of Terminal Services to enable persistent access to systems. Microsoft Terminal Services, renamed to Remote Desktop Services in some Windows Server OSs as of 2022, enable remote terminal connections to hosts. Terminal Services allows servers to transmit a full, interactive, graphical user interface to clients via RDP.
T1205	Traffic Signaling	Adversaries may use traffic signaling to hide open ports or other malicious functionality used for persistence or command and control. Traffic signaling involves the use of a magic value or sequence that must be sent to a system to trigger a special response, such as opening a closed port or executing a malicious task. This may take the form of sending a series of packets with certain characteristics before a port will be opened that the adversary can use for command and control. Usually this series of packets consists of attempted connections to a predefined sequence of closed ports (i.e. Port Knocking), but can involve unusual flags, specific strings, or other unique characteristics. After the sequence is completed, opening a port may be accomplished by the host-based firewall, but could also be implemented by custom software.
T1205.001	Port Knocking	Adversaries may use port knocking to hide open ports used for persistence or command and control. To enable a port, an adversary sends a series of attempted connections to a predefined sequence of closed ports. After the sequence is completed, opening a port is often accomplished by the host based firewall, but could also be implemented by custom software.

T1205.002	Socket Filters	Adversaries may attach filters to a network socket to monitor then activate backdoors used for persistence or command and control. With elevated permissions, adversaries can use features such as the libpcap library to open sockets and install filters to allow or disallow certain types of data to come through the socket. The filter may apply to all traffic passing through the specified network interface (or every interface if not specified). When the network interface receives a packet matching the filter criteria, additional actions can be triggered on the host, such as activation of a reverse shell.
T1078	Valid Accounts	Adversaries may obtain and abuse credentials of existing accounts as a means of gaining Initial Access, Persistence, Privilege Escalation, or Defense Evasion. Compromised credentials may be used to bypass access controls placed on various resources on systems within the network and may even be used for persistent access to remote systems and externally available services, such as VPNs, Outlook Web Access, network devices, 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.
T1078.001	Default Accounts	Adversaries may obtain and abuse credentials of a default account as a means of gaining Initial Access, Persistence, Privilege Escalation, or Defense Evasion. Default accounts are those that are built-into an OS, such as the Guest or Administrator accounts on Windows systems. Default accounts also include default factory/provider set accounts on other types of systems, software, or devices, including the root user account in AWS and the default service account in Kubernetes.

T1078.002	Domain Accounts	Adversaries may obtain and abuse credentials of a domain account as a means of gaining Initial Access, Persistence, Privilege Escalation, or Defense Evasion. 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.
T1078.003	Local Accounts	Adversaries may obtain and abuse credentials of a local account as a means of gaining Initial Access, Persistence, Privilege Escalation, or Defense Evasion. Local accounts are those configured by an organization for use by users, remote support, services, or for administration on a single system or service.
T1078.004	Cloud Accounts	Valid accounts in cloud environments may allow adversaries to perform actions to achieve Initial Access, Persistence, Privilege Escalation, or Defense Evasion. Cloud accounts are those created and configured by an organization for use by users, remote support, services, or for administration of resources within a cloud service provider or SaaS application. Cloud Accounts can exist solely in the cloud; alternatively, they may be hybrid-joined between on-premises systems and the cloud through syncing or federation with other identity sources such as Windows Active Directory.

Tactic: Privilege Escalation
ID: TA0004
Description:
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:

SYSTEM/root level

local administrator
user account with admin-like access
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.

Techniques Under the Privilege Escalation Tactic

ID	Name	Description
T1548	Abuse Elevation Control Mechanism	Adversaries may circumvent mechanisms designed to control elevate privileges to gain higher-level permissions. Most modern systems contain native elevation control mechanisms that are intended to limit privileges that a user can perform on a machine. Authorization has to be granted to specific users in order to perform tasks that can be considered of higher risk. An adversary can perform several methods to take advantage of built-in control mechanisms in order to escalate privileges on a system.
T1548.001	Setuid and Setgid	An adversary may abuse configurations where an application has the setuid or setgid bits set in order to get code running in a different (and possibly more privileged) user’s context. On Linux or macOS, when the setuid or setgid bits are set for an application binary, the application will run with the privileges of the owning user or group respectively. Normally an application is run in the current user’s context, regardless of which user or group owns the application. However, there are instances where programs need to be executed in an elevated context to function properly, but the user running them may not have the specific required privileges.

T1548.002	Bypass User Account Control	Adversaries may bypass UAC mechanisms to elevate process privileges on system. Windows User Account Control (UAC) allows a program to elevate its privileges (tracked as integrity levels ranging from low to high) to perform a task under administrator-level permissions, possibly 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.
T1548.003	Sudo and Sudo Caching	Adversaries may perform sudo caching and/or use the sudoers file to elevate privileges. Adversaries may do this to execute commands as other users or spawn processes with higher privileges.
T1548.004	Elevated Execution with Prompt	Adversaries may leverage the AuthorizationExecuteWithPrivileges API to escalate privileges by prompting the user for credentials. 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.
T1548.005	Temporary Elevated Cloud Access	Adversaries may abuse permission configurations that allow them to gain temporarily elevated access to cloud resources. Many cloud environments allow administrators to grant user or service accounts permission to request just-in-time access to roles, impersonate other accounts, pass roles onto resources and services, or otherwise gain short-term access to a set of privileges that may be distinct from their own.
T1548.006	TCC Manipulation	Adversaries can manipulate or abuse the Transparency, Consent, & Control (TCC) service or database to grant malicious executables elevated permissions. TCC is a Privacy & Security macOS control mechanism used to determine if the running process has permission to access the data or services protected by TCC, such as screen sharing, camera, microphone, or Full Disk Access (FDA).

T1134	Access Token Manipulation	Adversaries may modify access tokens to operate under a different user or system security context to perform actions and bypass access controls. 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 is the child of a different process or 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.
T1134.001	Token Impersonation/Theft	Adversaries may duplicate then impersonate another user's existing token to escalate privileges and bypass access controls. For example, an adversary can duplicate an existing token using DuplicateToken or DuplicateTokenEx. The token can then be used with ImpersonateLoggedOnUser to allow the calling thread to impersonate a logged on user's security context, or with SetThreadToken to assign the impersonated token to a thread.
T1134.002	Create Process with Token	Adversaries may create a new process with an existing token to escalate privileges and bypass access controls. Processes can be created with the token and resulting security context of another user using features such as CreateProcessWithTokenW and runas.
T1134.003	Make and Impersonate Token	Adversaries may make new tokens and impersonate users to escalate privileges and bypass access controls. For example, if 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 LogonUser function. The function will return a copy of the new session's access token and the adversary can use SetThreadToken to assign the token to a thread.

T1134.004	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 CreateProcess API call, which supports a parameter that defines the PPID to use. 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 svchost.exe or consent.exe) rather than the current user context.
T1134.005	SID-History Injection	Adversaries may use SID-History Injection to escalate privileges and bypass access controls. 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. An account can hold additional SIDs in the SID-History Active Directory attribute , allowing inter-operable account migration between domains (e.g., all values in SID-History are included in access tokens).
T1098	Account Manipulation	Adversaries may manipulate accounts to maintain and/or elevate access to victim systems. Account manipulation may consist of any action that preserves or modifies adversary access to a compromised account, such as modifying credentials or permission groups. These actions could also include account activity designed to subvert security policies, such as performing iterative password updates to bypass password duration policies and preserve the life of compromised credentials.
T1098.001	Additional Cloud Credentials	Adversaries may add adversary-controlled credentials to a cloud account to maintain persistent access to victim accounts and instances within the environment.
T1098.002	Additional Email Delegate Permissions	Adversaries may grant additional permission levels to maintain persistent access to an adversary-controlled email account.

T1098.003	Additional Cloud Roles	An adversary may add additional roles or permissions to an adversary-controlled cloud account to maintain persistent access to a tenant. For example, adversaries may update IAM policies in cloud-based environments or add a new global administrator in Office 365 environments. With sufficient permissions, a compromised account can gain almost unlimited access to data and settings (including the ability to reset the passwords of other admins).
T1098.004	SSH Authorized Keys	Adversaries may modify the SSH authorized_keys file to maintain persistence on a victim host. Linux distributions and macOS commonly use key-based authentication to secure the authentication process of SSH sessions for remote management. The authorized_keys file in SSH specifies the SSH keys that can be used for logging into the user account for which the file is configured. This file is usually found in the user's home directory under <user-home>/.ssh/authorized_keys. Users may edit the system’s SSH config file to modify the directives PubkeyAuthentication and RSAAuthentication to the value "yes" to ensure public key and RSA authentication are enabled. The SSH config file is usually located under /etc/ssh/sshd_config.
T1098.005	Device Registration	Adversaries may register a device to an adversary-controlled account. Devices may be registered in a multifactor authentication (MFA) system, which handles authentication to the network, or in a device management system, which handles device access and compliance.
T1098.006	Additional Container Cluster Roles	An adversary may add additional roles or permissions to an adversary-controlled user or service account to maintain persistent access to a container orchestration system. For example, an adversary with sufficient permissions may create a RoleBinding or a ClusterRoleBinding to bind a Role or ClusterRole to a Kubernetes account. Where attribute-based access control (ABAC) is in use, an adversary with sufficient permissions may modify a Kubernetes ABAC policy to give the target account additional permissions.

T1098.007	Additional Local or Domain Groups	An adversary may add additional local or domain groups to an adversary-controlled account to maintain persistent access to a system or domain.
T1547	Boot or Logon Autostart Execution	Adversaries may configure system settings to automatically execute a program during system boot or logon to maintain persistence or gain higher-level privileges on compromised systems. Operating systems may have mechanisms for automatically running a program on system boot or account logon. These mechanisms may include automatically executing programs that are placed in specially designated directories or are referenced by repositories that store configuration information, such as the Windows Registry. An adversary may achieve the same goal by modifying or extending features of the kernel.
T1547.001	Registry Run Keys / Startup Folder	Adversaries may achieve persistence by adding a program to a startup folder or referencing it with a Registry run key. Adding an entry to the "run keys" in the Registry or startup folder will cause the program referenced to be executed when a user logs in. These programs will be executed under the context of the user and will have the account's associated permissions level.
T1547.002	Authentication Package	Adversaries may abuse authentication packages to execute DLLs when the system boots. 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.
T1547.003	Time Providers	Adversaries may abuse time providers to execute DLLs when the system boots. The Windows Time service (W32Time) enables time synchronization across and within domains. W32Time time providers are responsible for retrieving time stamps from hardware/network resources and outputting these values to other network clients.

T1547.004	Winlogon Helper DLL	Adversaries may abuse features of Winlogon to execute DLLs and/or executables when a user logs in. 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 HKLM\Software[\Wow6432Node\]\Microsoft\Windows NT\CurrentVersion\Winlogon\ and HKCU\Software\Microsoft\Windows NT\CurrentVersion\Winlogon\ are used to manage additional helper programs and functionalities that support Winlogon.
T1547.005	Security Support Provider	Adversaries may abuse security support providers (SSPs) to execute DLLs when the system boots. Windows 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.
T1547.006	Kernel Modules and Extensions	Adversaries may modify the kernel to automatically execute programs on system boot. Loadable Kernel Modules (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.
T1547.007	Re-opened Applications	Adversaries may modify plist files to automatically run an application when a user logs in. When a user logs out or restarts via the macOS Graphical User Interface (GUI), a prompt is provided to the user with a checkbox to "Reopen windows when logging back in". When selected, all applications currently open are added to a property list file named com.apple.loginwindow.[UUID].plist within the ~/Library/Preferences/ByHost directory. Applications listed in this file are automatically reopened upon the user’s next logon.

T1547.008	LSASS Driver	Adversaries may modify or add LSASS drivers to obtain persistence on compromised systems. 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.
T1547.009	Shortcut Modification	Adversaries may create or modify shortcuts that can execute a program during system boot or user login. Shortcuts or symbolic links are used to reference other files or programs that will be opened or executed when the shortcut is clicked or executed by a system startup process.
T1547.01	Port Monitors	Adversaries may use port monitors to run an adversary supplied DLL during system boot for persistence or privilege escalation. A port monitor can be set through the AddMonitor API call to set a DLL to be loaded at startup. This DLL can be located in C:\Windows\System32 and will be loaded and run by the print spooler service, spoolsv.exe, under SYSTEM level permissions on boot.
T1547.012	Print Processors	Adversaries may abuse print processors to run malicious DLLs during system boot for persistence and/or privilege escalation. Print processors are DLLs that are loaded by the print spooler service, spoolsv.exe, during boot.
T1547.013	XDG Autostart Entries	Adversaries may add or modify XDG Autostart Entries to execute malicious programs or commands when a user’s desktop environment is loaded at login. XDG Autostart entries are available for any XDG-compliant Linux system. XDG Autostart entries use Desktop Entry files (.desktop) to configure the user’s desktop environment upon user login. These configuration files determine what applications launch upon user login, define associated applications to open specific file types, and define applications used to open removable media.

T1547.014	Active Setup	Adversaries may achieve persistence by adding a Registry key to the Active Setup of the local machine. Active Setup is a Windows mechanism that is used to execute programs when a user logs in. The value stored in the Registry key will be executed after a user logs into the computer. These programs will be executed under the context of the user and will have the account's associated permissions level.
T1547.015	Login Items	Adversaries may add login items to execute upon user login to gain persistence or escalate privileges. Login items are applications, documents, folders, or server connections that are automatically launched when a user logs in. Login items can be added via a shared file list or Service Management Framework. Shared file list login items can be set using scripting languages such as AppleScript, whereas the Service Management Framework uses the API call SMLoginItemSetEnabled.
T1037	Boot or Logon Initialization Scripts	Adversaries may use scripts automatically executed at boot or logon initialization to establish persistence. Initialization scripts can be used to perform administrative functions, which may often execute other programs or send information to an internal logging server. These scripts can vary based on operating system and whether applied locally or remotely.
T1037.001	Logon Script (Windows)	Adversaries may use Windows logon scripts automatically executed at logon initialization to establish persistence. Windows allows logon scripts to be run whenever a specific user or group of users log into a system. This is done via adding a path to a script to the HKCU\Environment\UserInitMprLogonScript Registry key.

T1037.002	Login Hook	Adversaries may use a Login Hook to establish persistence executed upon user logon. A login hook is a plist file that points to a specific script to execute with root privileges upon user logon. The plist file is located in the /Library/Preferences/com.apple.loginwindow.plist file and can be modified using the defaults command-line utility. This behavior is the same for logout hooks where a script can be executed upon user logout. All hooks require administrator permissions to modify or create hooks.
T1037.003	Network Logon Script	Adversaries may use network logon scripts automatically executed at logon initialization to establish persistence. Network logon scripts can be assigned using Active Directory or Group Policy Objects. These logon scripts run with the privileges of the user they are assigned to. Depending on the systems within the network, initializing one of these scripts could apply to more than one or potentially all systems.
T1037.004	RC Scripts	Adversaries may establish persistence by modifying RC scripts which are executed during a Unix-like system's startup. These files allow system administrators to map and start custom services at startup for different run levels. RC scripts require root privileges to modify.
T1037.005	Startup Items	Adversaries may use startup items automatically executed at boot initialization to establish persistence. 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.
T1543	Create or Modify System Process	Adversaries may create or modify system-level processes to repeatedly execute malicious payloads as part of persistence. When operating systems boot up, they can start processes that perform background system functions. On Windows and Linux, these system processes are referred to as services. On macOS, launchd processes known as Launch Daemon and Launch Agent are run to finish system initialization and load user specific parameters.

T1543.001	Launch Agent	Adversaries may create or modify launch agents to repeatedly execute malicious payloads as part of persistence. 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) file found in /System/Library/LaunchAgents, /Library/LaunchAgents, and ~/Library/LaunchAgents. Property list files use the Label, ProgramArguments , and RunAtLoad keys to identify the Launch Agent's name, executable location, and execution time. Launch Agents are often installed to perform updates to programs, launch user specified programs at login, or to conduct other developer tasks.
T1543.002	Systemd Service	Adversaries may create or modify systemd services to repeatedly execute malicious payloads as part of persistence. Systemd is a system and service manager commonly used for managing background daemon processes (also known as services) and other system resources. Systemd is the default initialization (init) system on many Linux distributions replacing legacy init systems, including SysVinit and Upstart, while remaining backwards compatible.
T1543.003	Windows Service	Adversaries may create or modify Windows services to repeatedly execute malicious payloads as part of persistence. When Windows boots up, it starts programs or applications called services that perform background system functions. Windows service configuration information, including the file path to the service's executable or recovery programs/commands, is stored in the Windows Registry.

T1543.004	Launch Daemon	Adversaries may create or modify Launch Daemons to execute malicious payloads as part of persistence. Launch Daemons are plist files used to interact with Launchd, the service management framework used by macOS. Launch Daemons require elevated privileges to install, are executed for every user on a system prior to login, and run in the background without the need for user interaction. During the macOS initialization startup, the launchd process loads the parameters for launch-on-demand system-level daemons from plist files found in /System/Library/LaunchDaemons/ and /Library/LaunchDaemons/. Required Launch Daemons parameters include a Label to identify the task, Program to provide a path to the executable, and RunAtLoad to specify when the task is run. Launch Daemons are often used to provide access to shared resources, updates to software, or conduct automation tasks.
T1543.005	Container Service	Adversaries may create or modify container or container cluster management tools that run as daemons, agents, or services on individual hosts. These include software for creating and managing individual containers, such as Docker and Podman, as well as container cluster node-level agents such as kubelet. By modifying these services, an adversary may be able to achieve persistence or escalate their privileges on a host.
T1484	Domain or Tenant Policy Modification	Adversaries may modify the configuration settings of a domain or identity tenant to evade defenses and/or escalate privileges in centrally managed environments. Such services provide a centralized means of managing identity resources such as devices and accounts, and often include configuration settings that may apply between domains or tenants such as trust relationships, identity syncing, or identity federation.
T1484.001	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 predictable network path \<DOMAIN>\SYSVOL\<DOMAIN>\Policies\.

T1484.002	Trust Modification	Adversaries may add new domain trusts, modify the properties of existing domain trusts, or otherwise change the configuration of trust relationships between domains and tenants to evade defenses and/or elevate privileges.Trust details, such as whether or not user identities are federated, allow authentication and authorization properties to apply between domains or tenants for the purpose of accessing shared resources. These trust objects may include accounts, credentials, and other authentication material applied to servers, tokens, and domains.
T1611	Escape to Host	Adversaries may break out of a container to gain access to the underlying host. This can allow an adversary access to other containerized resources from the host level or to the host itself. In principle, containerized resources should provide a clear separation of application functionality and be isolated from the host environment.
T1546	Event Triggered Execution	Adversaries may establish persistence and/or elevate privileges using system mechanisms that trigger execution based on specific events. Various operating systems have means to monitor and subscribe to events such as logons or other user activity such as running specific applications/binaries. Cloud environments may also support various functions and services that monitor and can be invoked in response to specific cloud events.
T1546.001	Change Default File Association	Adversaries may establish persistence by executing malicious content triggered by a file type 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 or by administrators using the built-in assoc utility. 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.

T1546.002	Screensaver	Adversaries may establish persistence by executing malicious content triggered by user inactivity. Screensavers are programs that execute after a configurable time of user inactivity and consist of Portable Executable (PE) files with a .scr file extension. The Windows screensaver application scrnsave.scr is located in C:\Windows\System32\, and C:\Windows\sysWOW64\ on 64-bit Windows systems, along with screensavers included with base Windows installations.
T1546.003	Windows Management Instrumentation Event Subscription	Adversaries may establish persistence and elevate privileges by executing malicious content triggered by a Windows Management Instrumentation (WMI) event subscription. WMI can be used to install event filters, providers, consumers, and bindings that execute code when a defined event occurs. Examples of events that may be subscribed to are the wall clock time, user login, or the computer's uptime.
T1546.004	Unix Shell Configuration Modification	Adversaries may establish persistence through executing malicious commands triggered by a user’s shell. User Unix Shells execute several configuration scripts at different points throughout the session based on events. For example, when a user opens a command-line interface or remotely logs in (such as via SSH) a login shell is initiated. The login shell executes scripts from the system (/etc) and the user’s home directory (~/) to configure the environment. All login shells on a system use /etc/profile when initiated. These configuration scripts run at the permission level of their directory and are often used to set environment variables, create aliases, and customize the user’s environment. When the shell exits or terminates, additional shell scripts are executed to ensure the shell exits appropriately.
T1546.005	Trap	Adversaries may establish persistence by executing malicious content triggered by an interrupt signal. The trap 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 ctrl+c and ctrl+d.

T1546.006	LC_LOAD_DYLIB Addition	Adversaries may establish persistence by executing malicious content triggered by the execution of tainted binaries. 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 as adjustments are made to the rest of the fields and dependencies. There are tools available to perform these changes.
T1546.007	Netsh Helper DLL	Adversaries may establish persistence by executing malicious content triggered by Netsh Helper DLLs. 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. The paths to registered netsh.exe helper DLLs are entered into the Windows Registry at HKLM\SOFTWARE\Microsoft\Netsh.
T1546.008	Accessibility Features	Adversaries may establish persistence and/or elevate privileges by executing malicious content triggered by accessibility features. Windows contains accessibility features that may be launched with a key combination before a user has logged in (ex: 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.
T1546.009	AppCert DLLs	Adversaries may establish persistence and/or elevate privileges by executing malicious content triggered by AppCert DLLs loaded into processes. Dynamic-link libraries (DLLs) that are specified in the AppCertDLLs Registry key under HKEY_LOCAL_MACHINE\System\CurrentControlSet\Control\Session Manager\ are loaded into every process that calls the ubiquitously used application programming interface (API) functions CreateProcess, CreateProcessAsUser, CreateProcessWithLoginW, CreateProcessWithTokenW, or WinExec.

T1546.01	AppInit DLLs	Adversaries may establish persistence and/or elevate privileges by executing malicious content triggered by AppInit DLLs loaded into processes. Dynamic-link libraries (DLLs) that are specified in the AppInit_DLLs value in the Registry keys HKEY_LOCAL_MACHINE\Software\Microsoft\Windows NT\CurrentVersion\Windows or HKEY_LOCAL_MACHINE\Software\Wow6432Node\Microsoft\Windows NT\CurrentVersion\Windows 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.
T1546.011	Application Shimming	Adversaries may establish persistence and/or elevate privileges by executing malicious content triggered by application shims. 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.
T1546.012	Image File Execution Options Injection	Adversaries may establish persistence and/or elevate privileges by executing malicious content triggered by Image File Execution Options (IFEO) debuggers. IFEOs 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).
T1546.013	PowerShell Profile	Adversaries may gain persistence and elevate privileges by executing malicious content triggered by PowerShell profiles. A PowerShell profile (profile.ps1) is a script that runs when PowerShell starts and can be used as a logon script to customize user environments.
T1546.014	Emond	Adversaries may gain persistence and elevate privileges by executing malicious content triggered by the Event Monitor Daemon (emond). Emond is a Launch Daemon that accepts events from various services, runs them through a simple rules engine, and takes action. The emond binary at /sbin/emond will load any rules from the /etc/emond.d/rules/ directory and take action once an explicitly defined event takes place.

T1546.015	Component Object Model Hijacking	Adversaries may establish persistence by executing malicious content triggered by hijacked references to Component Object Model (COM) objects. COM is a system within Windows to enable interaction between software components through the operating system. References to various COM objects are stored in the Registry.
T1546.016	Installer Packages	Adversaries may establish persistence and elevate privileges by using an installer to trigger the execution of malicious content. Installer packages are OS specific and contain the resources an operating system needs to install applications on a system. Installer packages can include scripts that run prior to installation as well as after installation is complete. Installer scripts may inherit elevated permissions when executed. Developers often use these scripts to prepare the environment for installation, check requirements, download dependencies, and remove files after installation.
T1546017	Udev Rules	Adversaries may maintain persistence through executing malicious content triggered using udev rules. Udev is the Linux kernel device manager that dynamically manages device nodes, handles access to pseudo-device files in the /dev directory, and responds to hardware events, such as when external devices like hard drives or keyboards are plugged in or removed. Udev uses rule files with match keys to specify the conditions a hardware event must meet and action keys to define the actions that should follow. Root permissions are required to create, modify, or delete rule files located in /etc/udev/rules.d/, /run/udev/rules.d/, /usr/lib/udev/rules.d/, /usr/local/lib/udev/rules.d/, and /lib/udev/rules.d/. Rule priority is determined by both directory and by the digit prefix in the rule filename.

T1068	Exploitation for Privilege Escalation	Adversaries may exploit software vulnerabilities in an attempt to elevate privileges. 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.
T1574	Hijack Execution Flow	Adversaries may execute their own malicious payloads by hijacking the way operating systems run programs. Hijacking execution flow can be for the purposes of persistence, since this hijacked execution may reoccur over time. Adversaries may also use these mechanisms to elevate privileges or evade defenses, such as application control or other restrictions on execution.
T1574.001	DLL Search Order Hijacking	Adversaries may execute their own malicious payloads by hijacking the search order used to load DLLs. Windows systems use a common method to look for required DLLs to load into a program. Hijacking DLL loads may be for the purpose of establishing persistence as well as elevating privileges and/or evading restrictions on file execution.
T1574.002	DLL Side-Loading	Adversaries may execute their own malicious payloads by side-loading DLLs. Similar to DLL Search Order Hijacking, side-loading involves hijacking which DLL a program loads. But rather than just planting the DLL within the search order of a program then waiting for the victim application to be invoked, adversaries may directly side-load their payloads by planting then invoking a legitimate application that executes their payload(s).

T1574.004	Dylib Hijacking	Adversaries may execute their own payloads by placing a malicious dynamic library (dylib) with an expected name in a path a victim application searches at runtime. The dynamic loader will try to find the dylibs based on the sequential order of the search paths. Paths to dylibs may be prefixed with @rpath, which allows developers to use relative paths to specify an array of search paths used at runtime based on the location of the executable. Additionally, if weak linking is used, such as the LC_LOAD_WEAK_DYLIB function, an application will still execute even if an expected dylib is not present. Weak linking enables developers to run an application on multiple macOS versions as new APIs are added.
T1574.005	Executable Installer File Permissions Weakness	Adversaries may execute their own malicious payloads by hijacking the binaries used by an installer. These 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.
T1574.006	Dynamic Linker Hijacking	Adversaries may execute their own malicious payloads by hijacking environment variables the dynamic linker uses to load shared libraries. During the execution preparation phase of a program, the dynamic linker loads specified absolute paths of shared libraries from environment variables and files, such as LD_PRELOAD on Linux or DYLD_INSERT_LIBRARIES on macOS. Libraries specified in environment variables are loaded first, taking precedence over system libraries with the same function name. These variables are often used by developers to debug binaries without needing to recompile, deconflict mapped symbols, and implement custom functions without changing the original library.

T1574.007	Path Interception by PATH Environment Variable	Adversaries may execute their own malicious payloads by hijacking environment variables used to load libraries. The PATH environment variable contains a list of directories (User and System) that the OS searches sequentially through in search of the binary that was called from a script or the command line.
T1574.008	Path Interception by Search Order Hijacking	Adversaries may execute their own malicious payloads by hijacking the search order used to load other programs. Because some programs do not call other programs using the full path, adversaries may place their own file in the directory where the calling program is located, causing the operating system to launch their malicious software at the request of the calling program.
T1574.009	Path Interception by Unquoted Path	Adversaries may execute their own malicious payloads by hijacking vulnerable file path references. Adversaries can take advantage of paths that lack surrounding quotations by placing an executable in a higher level directory within the path, so that Windows will choose the adversary's executable to launch.
T1574.01	Services File Permissions Weakness	Adversaries may execute their own malicious payloads by hijacking the binaries used by services. Adversaries may use flaws in the permissions of Windows services to replace the binary that is executed upon service start. These service 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.

T1574.011	Services Registry Permissions Weakness	Adversaries may execute their own malicious payloads by hijacking the Registry entries used by services. Adversaries may use flaws in the permissions for Registry keys related to services to redirect from the originally specified executable to one that they control, in order to launch their own code when a service starts. Windows stores local service configuration information in the Registry under HKLM\SYSTEM\CurrentControlSet\Services. 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, or Reg. Access to Registry keys is controlled through access control lists and user permissions.
T1574.012	COR_PROFILER	Adversaries may leverage the COR_PROFILER environment variable to hijack the execution flow of programs that load the .NET CLR. The COR_PROFILER is a .NET Framework feature which allows developers to specify an unmanaged (or external of .NET) profiling DLL to be loaded into each .NET process that loads the Common Language Runtime (CLR). These profilers are designed to monitor, troubleshoot, and debug managed code executed by the .NET CLR.
T1574.013	KernelCallbackTable	Adversaries may abuse the KernelCallbackTable of a process to hijack its execution flow in order to run their own payloads. The KernelCallbackTable can be found in the Process Environment Block (PEB) and is initialized to an array of graphic functions available to a GUI process once user32.dll is loaded.
T1574.014	AppDomainManager	Adversaries may execute their own malicious payloads by hijacking how the .NET AppDomainManager loads assemblies. The .NET framework uses the AppDomainManager class to create and manage one or more isolated runtime environments (called application domains) inside a process to host the execution of .NET applications. Assemblies (.exe or .dll binaries compiled to run as .NET code) may be loaded into an application domain as executable code.

T1055	Process Injection	Adversaries may inject code into processes in order to evade process-based defenses as well as possibly elevate privileges. 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.
T1055.001	Dynamic-link Library Injection	Adversaries may inject dynamic-link libraries (DLLs) into processes in order to evade process-based defenses as well as possibly elevate privileges. DLL injection is a method of executing arbitrary code in the address space of a separate live process.
T1055.002	Portable Executable Injection	Adversaries may inject portable executables (PE) into processes in order to evade process-based defenses as well as possibly elevate privileges. PE injection is a method of executing arbitrary code in the address space of a separate live process.
T1055.003	Thread Execution Hijacking	Adversaries may inject malicious code into hijacked processes in order to evade process-based defenses as well as possibly elevate privileges. Thread Execution Hijacking is a method of executing arbitrary code in the address space of a separate live process.
T1055.004	Asynchronous Procedure Call	Adversaries may inject malicious code into processes via the asynchronous procedure call (APC) queue in order to evade process-based defenses as well as possibly elevate privileges. APC injection is a method of executing arbitrary code in the address space of a separate live process.
T1055.005	Thread Local Storage	Adversaries may inject malicious code into processes via thread local storage (TLS) callbacks in order to evade process-based defenses as well as possibly elevate privileges. TLS callback injection is a method of executing arbitrary code in the address space of a separate live process.

T1055.008	Ptrace System Calls	Adversaries may inject malicious code into processes via ptrace (process trace) system calls in order to evade process-based defenses as well as possibly elevate privileges. Ptrace system call injection is a method of executing arbitrary code in the address space of a separate live process.
T1055.009	Proc Memory	Adversaries may inject malicious code into processes via the /proc filesystem in order to evade process-based defenses as well as possibly elevate privileges. Proc memory injection is a method of executing arbitrary code in the address space of a separate live process.
T1055.011	Extra Window Memory Injection	Adversaries may inject malicious code into process via Extra Window Memory (EWM) in order to evade process-based defenses as well as possibly elevate privileges. EWM injection is a method of executing arbitrary code in the address space of a separate live process.
T1055.012	Process Hollowing	Adversaries may inject malicious code into suspended and hollowed processes in order to evade process-based defenses. Process hollowing is a method of executing arbitrary code in the address space of a separate live process.
T1055.013	Process Doppelgänging	Adversaries may inject malicious code into process via process doppelgänging in order to evade process-based defenses as well as possibly elevate privileges. Process doppelgänging is a method of executing arbitrary code in the address space of a separate live process.
T1055.014	VDSO Hijacking	Adversaries may inject malicious code into processes via VDSO hijacking in order to evade process-based defenses as well as possibly elevate privileges. Virtual dynamic shared object (vdso) hijacking is a method of executing arbitrary code in the address space of a separate live process.

T1055.015	ListPlanting	Adversaries may abuse list-view controls to inject malicious code into hijacked processes in order to evade process-based defenses as well as possibly elevate privileges. ListPlanting is a method of executing arbitrary code in the address space of a separate live process. Code executed via ListPlanting may also evade detection from security products since the execution is masked under a legitimate process.
T1053	Scheduled Task/Job	Adversaries may abuse task scheduling functionality to facilitate initial or recurring execution of malicious code. Utilities exist within all major operating systems to schedule programs or scripts to be executed at a specified date and time. A task can also be scheduled on a remote system, provided the proper authentication is met (ex: RPC and file and printer sharing in Windows environments). Scheduling a task on a remote system typically may require being a member of an admin or otherwise privileged group on the remote system.
T1053.002	At	Adversaries may abuse the at utility to perform task scheduling for initial or recurring execution of malicious code. The at utility exists as an executable within Windows, Linux, and macOS for scheduling tasks at a specified time and date. Although deprecated in favor of Scheduled Task's schtasks in Windows environments, using at requires that the Task Scheduler service be running, and the user to be logged on as a member of the local Administrators group. In addition to explicitly running the at command, adversaries may also schedule a task with at by directly leveraging the Windows Management Instrumentation Win32_ScheduledJob WMI class.
T1053.003	Cron	Adversaries may abuse the cron utility to perform task scheduling for initial or recurring execution of malicious code. The cron utility is a time-based job scheduler for Unix-like operating systems. The crontab file contains the schedule of cron entries to be run and the specified times for execution. Any crontab files are stored in operating system-specific file paths.

T1053.005	Scheduled Task	Adversaries may abuse the Windows Task Scheduler to perform task scheduling for initial or recurring execution of malicious code. There are multiple ways to access the Task Scheduler in Windows. The schtasks utility can be run directly on the command line, or the Task Scheduler can be opened through the GUI within the Administrator Tools section of the Control Panel. In some cases, adversaries have used a .NET wrapper for the Windows Task Scheduler, and alternatively, adversaries have used the Windows netapi32 library and Windows Management Instrumentation (WMI) to create a scheduled task. Adversaries may also utilize the Powershell Cmdlet Invoke-CimMethod, which leverages WMI class PS_ScheduledTask to create a scheduled task via an XML path.
T1053.006	Systemd Timers	Adversaries may abuse systemd timers to perform task scheduling for initial or recurring execution of malicious code. Systemd timers are unit files with file extension .timer that control services. Timers can be set to run on a calendar event or after a time span relative to a starting point. They can be used as an alternative to Cron in Linux environments. Systemd timers may be activated remotely via the systemctl command line utility, which operates over SSH.
T1053.007	Container Orchestration Job	Adversaries may abuse task scheduling functionality provided by container orchestration tools such as Kubernetes to schedule deployment of containers configured to execute malicious code. Container orchestration jobs run these automated tasks at a specific date and time, similar to cron jobs on a Linux system. Deployments of this type can also be configured to maintain a quantity of containers over time, automating the process of maintaining persistence within a cluster.

T1078	Valid Accounts	Adversaries may obtain and abuse credentials of existing accounts as a means of gaining Initial Access, Persistence, Privilege Escalation, or Defense Evasion. Compromised credentials may be used to bypass access controls placed on various resources on systems within the network and may even be used for persistent access to remote systems and externally available services, such as VPNs, Outlook Web Access, network devices, 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.
T1078.001	Default Accounts	Adversaries may obtain and abuse credentials of a default account as a means of gaining Initial Access, Persistence, Privilege Escalation, or Defense Evasion. Default accounts are those that are built-into an OS, such as the Guest or Administrator accounts on Windows systems. Default accounts also include default factory/provider set accounts on other types of systems, software, or devices, including the root user account in AWS and the default service account in Kubernetes.
T1078.002	Domain Accounts	Adversaries may obtain and abuse credentials of a domain account as a means of gaining Initial Access, Persistence, Privilege Escalation, or Defense Evasion. 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.
T1078.003	Local Accounts	Adversaries may obtain and abuse credentials of a local account as a means of gaining Initial Access, Persistence, Privilege Escalation, or Defense Evasion. Local accounts are those configured by an organization for use by users, remote support, services, or for administration on a single system or service.

T1078.004	Cloud Accounts	Valid accounts in cloud environments may allow adversaries to perform actions to achieve Initial Access, Persistence, Privilege Escalation, or Defense Evasion. Cloud accounts are those created and configured by an organization for use by users, remote support, services, or for administration of resources within a cloud service provider or SaaS application. Cloud Accounts can exist solely in the cloud; alternatively, they may be hybrid-joined between on-premises systems and the cloud through syncing or federation with other identity sources such as Windows Active Directory.

Tactic: Defense Evasion
ID: TA0005
Description:
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.

Techniques Under the Defense Evasion Tactic

ID	Name	Description
T1548	Abuse Elevation Control Mechanism	Adversaries may circumvent mechanisms designed to control elevate privileges to gain higher-level permissions. Most modern systems contain native elevation control mechanisms that are intended to limit privileges that a user can perform on a machine. Authorization has to be granted to specific users in order to perform tasks that can be considered of higher risk. An adversary can perform several methods to take advantage of built-in control mechanisms in order to escalate privileges on a system.

T1548.001	Setuid and Setgid	An adversary may abuse configurations where an application has the setuid or setgid bits set in order to get code running in a different (and possibly more privileged) user’s context. On Linux or macOS, when the setuid or setgid bits are set for an application binary, the application will run with the privileges of the owning user or group respectively. Normally an application is run in the current user’s context, regardless of which user or group owns the application. However, there are instances where programs need to be executed in an elevated context to function properly, but the user running them may not have the specific required privileges.
T1548.002	Bypass User Account Control	Adversaries may bypass UAC mechanisms to elevate process privileges on system. Windows User Account Control (UAC) allows a program to elevate its privileges (tracked as integrity levels ranging from low to high) to perform a task under administrator-level permissions, possibly 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.
T1548.003	Sudo and Sudo Caching	Adversaries may perform sudo caching and/or use the sudoers file to elevate privileges. Adversaries may do this to execute commands as other users or spawn processes with higher privileges.
T1548.004	Elevated Execution with Prompt	Adversaries may leverage the AuthorizationExecuteWithPrivileges API to escalate privileges by prompting the user for credentials. 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.

T1548.005	Temporary Elevated Cloud Access	Adversaries may abuse permission configurations that allow them to gain temporarily elevated access to cloud resources. Many cloud environments allow administrators to grant user or service accounts permission to request just-in-time access to roles, impersonate other accounts, pass roles onto resources and services, or otherwise gain short-term access to a set of privileges that may be distinct from their own.
T1548.006	TCC Manipulation	Adversaries can manipulate or abuse the Transparency, Consent, & Control (TCC) service or database to grant malicious executables elevated permissions. TCC is a Privacy & Security macOS control mechanism used to determine if the running process has permission to access the data or services protected by TCC, such as screen sharing, camera, microphone, or Full Disk Access (FDA).
T1134	Access Token Manipulation	Adversaries may modify access tokens to operate under a different user or system security context to perform actions and bypass access controls. 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 is the child of a different process or 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.
T1134.001	Token Impersonation/Theft	Adversaries may duplicate then impersonate another user's existing token to escalate privileges and bypass access controls. For example, an adversary can duplicate an existing token using DuplicateToken or DuplicateTokenEx. The token can then be used with ImpersonateLoggedOnUser to allow the calling thread to impersonate a logged on user's security context, or with SetThreadToken to assign the impersonated token to a thread.

T1134.002	Create Process with Token	Adversaries may create a new process with an existing token to escalate privileges and bypass access controls. Processes can be created with the token and resulting security context of another user using features such as CreateProcessWithTokenW and runas.
T1134.003	Make and Impersonate Token	Adversaries may make new tokens and impersonate users to escalate privileges and bypass access controls. For example, if 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 LogonUser function. The function will return a copy of the new session's access token and the adversary can use SetThreadToken to assign the token to a thread.
T1134.004	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 CreateProcess API call, which supports a parameter that defines the PPID to use. 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 svchost.exe or consent.exe) rather than the current user context.
T1134.005	SID-History Injection	Adversaries may use SID-History Injection to escalate privileges and bypass access controls. 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. An account can hold additional SIDs in the SID-History Active Directory attribute , allowing inter-operable account migration between domains (e.g., all values in SID-History are included in access tokens).

T1197	BITS Jobs	Adversaries may abuse BITS jobs to persistently execute code and perform various background tasks. Windows Background Intelligent Transfer Service (BITS) is a low-bandwidth, asynchronous file transfer mechanism exposed through Component Object Model (COM). 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.
T1612	Build Image on Host	Adversaries may build a container image directly on a host to bypass defenses that monitor for the retrieval of malicious images from a public registry. A remote build request may be sent to the Docker API that includes a Dockerfile that pulls a vanilla base image, such as alpine, from a public or local registry and then builds a custom image upon it.
T1622	Debugger Evasion	Adversaries may employ various means to detect and avoid debuggers. Debuggers are typically used by defenders to trace and/or analyze the execution of potential malware payloads.
T1140	Deobfuscate/Decode Files or Information	Adversaries may use Obfuscated Files or Information to hide artifacts of an intrusion from analysis. They may require separate mechanisms to decode or deobfuscate that information depending on how they intend to use it. Methods for doing that include built-in functionality of malware or by using utilities present on the system.

T1610	Deploy Container	Adversaries may deploy a container into an environment to facilitate execution or evade defenses. In some cases, adversaries may deploy a new container to execute processes associated with a particular image or deployment, such as processes that execute or download malware. In others, an adversary may deploy a new container configured without network rules, user limitations, etc. to bypass existing defenses within the environment. In Kubernetes environments, an adversary may attempt to deploy a privileged or vulnerable container into a specific node in order to Escape to Host and access other containers running on the node.
T1006	Direct Volume Access	Adversaries may directly access a volume to bypass file access controls and file system monitoring. 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 may bypass Windows file access controls as well as file system monitoring tools.
T1484	Domain or Tenant Policy Modification	Adversaries may modify the configuration settings of a domain or identity tenant to evade defenses and/or escalate privileges in centrally managed environments. Such services provide a centralized means of managing identity resources such as devices and accounts, and often include configuration settings that may apply between domains or tenants such as trust relationships, identity syncing, or identity federation.
T1484.001	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 predictable network path \<DOMAIN>\SYSVOL\<DOMAIN>\Policies\.

T1484.002	Trust Modification	Adversaries may add new domain trusts, modify the properties of existing domain trusts, or otherwise change the configuration of trust relationships between domains and tenants to evade defenses and/or elevate privileges.Trust details, such as whether or not user identities are federated, allow authentication and authorization properties to apply between domains or tenants for the purpose of accessing shared resources. These trust objects may include accounts, credentials, and other authentication material applied to servers, tokens, and domains.
T1480	Execution Guardrails	Adversaries may use execution guardrails to constrain execution or actions based on adversary supplied and 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. 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.
T1480.001	Environmental Keying	Adversaries may environmentally key payloads or other features of malware to evade defenses and constraint execution to a specific target environment. Environmental keying uses cryptography to constrain execution or actions based on adversary supplied environment specific conditions that are expected to be present on the target. Environmental keying is an implementation of Execution Guardrails that utilizes cryptographic techniques for deriving encryption/decryption keys from specific types of values in a given computing environment.
T1480.002	Mutual Exclusion	Adversaries may constrain execution or actions based on the presence of a mutex associated with malware. A mutex is a locking mechanism used to synchronize access to a resource. Only one thread or process can acquire a mutex at a given time.

T1211	Exploitation for Defense Evasion	Adversaries may exploit a system or application vulnerability to bypass security features. Exploitation of a 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. Vulnerabilities may exist in defensive security software that can be used to disable or circumvent them.
T1222	File and Directory Permissions Modification	Adversaries may modify file or directory permissions/attributes to evade access control lists (ACLs) and access protected files. File and directory permissions are commonly managed by ACLs configured by the file or directory owner, or users with the appropriate permissions. File and directory ACL implementations vary by platform, but generally explicitly designate which users or groups can perform which actions (read, write, execute, etc.).
T1222.001	Windows File and Directory Permissions Modification	Adversaries may modify file or directory permissions/attributes to evade access control lists (ACLs) and access protected files. File and directory permissions are commonly managed by ACLs configured by the file or directory owner, or users with the appropriate permissions. File and directory ACL implementations vary by platform, but generally explicitly designate which users or groups can perform which actions (read, write, execute, etc.).
T1222.002	Linux and Mac File and Directory Permissions Modification	Adversaries may modify file or directory permissions/attributes to evade access control lists (ACLs) and access protected files. File and directory permissions are commonly managed by ACLs configured by the file or directory owner, or users with the appropriate permissions. File and directory ACL implementations vary by platform, but generally explicitly designate which users or groups can perform which actions (read, write, execute, etc.).

T1564	Hide Artifacts	Adversaries may attempt to hide artifacts associated with their behaviors to evade detection. Operating systems may have features to hide various artifacts, such as important system files and administrative task execution, to avoid disrupting user work environments and prevent users from changing files or features on the system. Adversaries may abuse these features to hide artifacts such as files, directories, user accounts, or other system activity to evade detection.
T1564.001	Hidden Files and Directories	Adversaries may set files and directories to be hidden to evade detection mechanisms. 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 (dir /a for Windows and ls –a for Linux and macOS).
T1564.002	Hidden Users	Adversaries may use hidden users to hide the presence of user accounts they create or modify. Administrators may want to hide users when there are many user accounts on a given system or if they want to hide their administrative or other management accounts from other users.
T1564.003	Hidden Window	Adversaries may use 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.

T1564.004	NTFS File Attributes	Adversaries may use NTFS file attributes to hide their malicious data in order to evade detection. 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. Within MFT entries are file attributes, 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).
T1564.005	Hidden File System	Adversaries may use a hidden file system to conceal malicious activity from users and security tools. File systems provide a structure to store and access data from physical storage. Typically, a user engages with a file system through applications that allow them to access files and directories, which are an abstraction from their physical location (ex: disk sector). Standard file systems include FAT, NTFS, ext4, and APFS. File systems can also contain other structures, such as the Volume Boot Record (VBR) and Master File Table (MFT) in NTFS.
T1564.006	Run Virtual Instance	Adversaries may carry out malicious operations using a virtual instance to avoid detection. A wide variety of virtualization technologies exist that allow for the emulation of a computer or computing environment. By running malicious code inside of a virtual instance, adversaries can hide artifacts associated with their behavior from security tools that are unable to monitor activity inside the virtual instance. Additionally, depending on the virtual networking implementation (ex: bridged adapter), network traffic generated by the virtual instance can be difficult to trace back to the compromised host as the IP address and hostname might not match known values.
T1564.007	VBA Stomping	Adversaries may hide malicious Visual Basic for Applications (VBA) payloads embedded within MS Office documents by replacing the VBA source code with benign data.

T1564.008	Email Hiding Rules	Adversaries may use email rules to hide inbound emails in a compromised user's mailbox. Many email clients allow users to create inbox rules for various email functions, including moving emails to other folders, marking emails as read, or deleting emails. Rules may be created or modified within email clients or through external features such as the New-InboxRule or Set-InboxRule PowerShell cmdlets on Windows systems.
T1564.009	Resource Forking	Adversaries may abuse resource forks to hide malicious code or executables to evade detection and bypass security applications. A resource fork provides applications a structured way to store resources such as thumbnail images, menu definitions, icons, dialog boxes, and code. Usage of a resource fork is identifiable when displaying a file's extended attributes, using ls -l@ or xattr -l commands. Resource forks have been deprecated and replaced with the application bundle structure. Non-localized resources are placed at the top level directory of an application bundle, while localized resources are placed in the /Resources folder.
T1564.01	Process Argument Spoofing	Adversaries may attempt to hide process command-line arguments by overwriting process memory. Process command-line arguments are stored in the process environment block (PEB), a data structure used by Windows to store various information about/used by a process. The PEB includes the process command-line arguments that are referenced when executing the process. When a process is created, defensive tools/sensors that monitor process creations may retrieve the process arguments from the PEB.

T1564.011	Ignore Process Interrupts	Adversaries may evade defensive mechanisms by executing commands that hide from process interrupt signals. Many operating systems use signals to deliver messages to control process behavior. Command interpreters often include specific commands/flags that ignore errors and other hangups, such as when the user of the active session logs off. These interrupt signals may also be used by defensive tools and/or analysts to pause or terminate specified running processes.
T1564.012	File/Path Exclusions	Adversaries may attempt to hide their file-based artifacts by writing them to specific folders or file names excluded from antivirus (AV) scanning and other defensive capabilities. AV and other file-based scanners often include exclusions to optimize performance as well as ease installation and legitimate use of applications. These exclusions may be contextual (e.g., scans are only initiated in response to specific triggering events/alerts), but are also often hardcoded strings referencing specific folders and/or files assumed to be trusted and legitimate.
T1574	Hijack Execution Flow	Adversaries may execute their own malicious payloads by hijacking the way operating systems run programs. Hijacking execution flow can be for the purposes of persistence, since this hijacked execution may reoccur over time. Adversaries may also use these mechanisms to elevate privileges or evade defenses, such as application control or other restrictions on execution.
T1574.001	DLL Search Order Hijacking	Adversaries may execute their own malicious payloads by hijacking the search order used to load DLLs. Windows systems use a common method to look for required DLLs to load into a program. Hijacking DLL loads may be for the purpose of establishing persistence as well as elevating privileges and/or evading restrictions on file execution.

T1574.002	DLL Side-Loading	Adversaries may execute their own malicious payloads by side-loading DLLs. Similar to DLL Search Order Hijacking, side-loading involves hijacking which DLL a program loads. But rather than just planting the DLL within the search order of a program then waiting for the victim application to be invoked, adversaries may directly side-load their payloads by planting then invoking a legitimate application that executes their payload(s).
T1574.004	Dylib Hijacking	Adversaries may execute their own payloads by placing a malicious dynamic library (dylib) with an expected name in a path a victim application searches at runtime. The dynamic loader will try to find the dylibs based on the sequential order of the search paths. Paths to dylibs may be prefixed with @rpath, which allows developers to use relative paths to specify an array of search paths used at runtime based on the location of the executable. Additionally, if weak linking is used, such as the LC_LOAD_WEAK_DYLIB function, an application will still execute even if an expected dylib is not present. Weak linking enables developers to run an application on multiple macOS versions as new APIs are added.
T1574.005	Executable Installer File Permissions Weakness	Adversaries may execute their own malicious payloads by hijacking the binaries used by an installer. These 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.

T1574.006	Dynamic Linker Hijacking	Adversaries may execute their own malicious payloads by hijacking environment variables the dynamic linker uses to load shared libraries. During the execution preparation phase of a program, the dynamic linker loads specified absolute paths of shared libraries from environment variables and files, such as LD_PRELOAD on Linux or DYLD_INSERT_LIBRARIES on macOS. Libraries specified in environment variables are loaded first, taking precedence over system libraries with the same function name. These variables are often used by developers to debug binaries without needing to recompile, deconflict mapped symbols, and implement custom functions without changing the original library.
T1574.007	Path Interception by PATH Environment Variable	Adversaries may execute their own malicious payloads by hijacking environment variables used to load libraries. The PATH environment variable contains a list of directories (User and System) that the OS searches sequentially through in search of the binary that was called from a script or the command line.
T1574.008	Path Interception by Search Order Hijacking	Adversaries may execute their own malicious payloads by hijacking the search order used to load other programs. Because some programs do not call other programs using the full path, adversaries may place their own file in the directory where the calling program is located, causing the operating system to launch their malicious software at the request of the calling program.
T1574.009	Path Interception by Unquoted Path	Adversaries may execute their own malicious payloads by hijacking vulnerable file path references. Adversaries can take advantage of paths that lack surrounding quotations by placing an executable in a higher level directory within the path, so that Windows will choose the adversary's executable to launch.

T1574.01	Services File Permissions Weakness	Adversaries may execute their own malicious payloads by hijacking the binaries used by services. Adversaries may use flaws in the permissions of Windows services to replace the binary that is executed upon service start. These service 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.
T1574.011	Services Registry Permissions Weakness	Adversaries may execute their own malicious payloads by hijacking the Registry entries used by services. Adversaries may use flaws in the permissions for Registry keys related to services to redirect from the originally specified executable to one that they control, in order to launch their own code when a service starts. Windows stores local service configuration information in the Registry under HKLM\SYSTEM\CurrentControlSet\Services. 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, or Reg. Access to Registry keys is controlled through access control lists and user permissions.
T1574.012	COR_PROFILER	Adversaries may leverage the COR_PROFILER environment variable to hijack the execution flow of programs that load the .NET CLR. The COR_PROFILER is a .NET Framework feature which allows developers to specify an unmanaged (or external of .NET) profiling DLL to be loaded into each .NET process that loads the Common Language Runtime (CLR). These profilers are designed to monitor, troubleshoot, and debug managed code executed by the .NET CLR.

T1574.013	KernelCallbackTable	Adversaries may abuse the KernelCallbackTable of a process to hijack its execution flow in order to run their own payloads. The KernelCallbackTable can be found in the Process Environment Block (PEB) and is initialized to an array of graphic functions available to a GUI process once user32.dll is loaded.
T1574.014	AppDomainManager	Adversaries may execute their own malicious payloads by hijacking how the .NET AppDomainManager loads assemblies. The .NET framework uses the AppDomainManager class to create and manage one or more isolated runtime environments (called application domains) inside a process to host the execution of .NET applications. Assemblies (.exe or .dll binaries compiled to run as .NET code) may be loaded into an application domain as executable code.
T1562	Impair Defenses	Adversaries may maliciously modify components of a victim environment in order to hinder or disable defensive mechanisms. This not only involves impairing preventative defenses, such as firewalls and anti-virus, but also detection capabilities that defenders can use to audit activity and identify malicious behavior. This may also span both native defenses as well as supplemental capabilities installed by users and administrators.
T1562.001	Disable or Modify Tools	Adversaries may modify and/or disable security tools to avoid possible detection of their malware/tools and activities. This may take many forms, such as killing security software processes or services, modifying / deleting Registry keys or configuration files so that tools do not operate properly, or other methods to interfere with security tools scanning or reporting information. Adversaries may also disable updates to prevent the latest security patches from reaching tools on victim systems.

T1562.002	Disable Windows Event Logging	Adversaries may disable Windows event logging to limit data that can be leveraged for detections and audits. Windows event logs record user and system activity such as login attempts, process creation, and much more. This data is used by security tools and analysts to generate detections.
T1562.003	Impair Command History Logging	Adversaries may impair command history logging to hide commands they run on a compromised system. Various command interpreters keep track of the commands users type in their terminal so that users can retrace what they've done.
T1562.004	Disable or Modify System Firewall	Adversaries may disable or modify system firewalls in order to bypass controls limiting network usage. Changes could be disabling the entire mechanism as well as adding, deleting, or modifying particular rules. This can be done numerous ways depending on the operating system, including via command-line, editing Windows Registry keys, and Windows Control Panel.
T1562.006	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 or even disabling host-based sensors, such as Event Tracing for Windows (ETW), by tampering settings that control the collection and flow of event telemetry. 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 or Windows Management Instrumentation.
T1562.007	Disable or Modify Cloud Firewall	Adversaries may disable or modify a firewall within a cloud environment to bypass controls that limit access to cloud resources. Cloud firewalls are separate from system firewalls that are described in Disable or Modify System Firewall.

T1562.008	Disable or Modify Cloud Logs	An adversary may disable or modify cloud logging capabilities and integrations to limit what data is collected on their activities and avoid detection. Cloud environments allow for collection and analysis of audit and application logs that provide insight into what activities a user does within the environment. If an adversary has sufficient permissions, they can disable or modify logging to avoid detection of their activities.
T1562.009	Safe Mode Boot	Adversaries may abuse Windows safe mode to disable endpoint defenses. Safe mode starts up the Windows operating system with a limited set of drivers and services. Third-party security software such as endpoint detection and response (EDR) tools may not start after booting Windows in safe mode. There are two versions of safe mode: Safe Mode and Safe Mode with Networking. It is possible to start additional services after a safe mode boot.
T1562.01	Downgrade Attack	Adversaries may downgrade or use a version of system features that may be outdated, vulnerable, and/or does not support updated security controls. Downgrade attacks typically take advantage of a system's backward compatibility to force it into less secure modes of operation.
T1562.011	Spoof Security Alerting	Adversaries may spoof security alerting from tools, presenting false evidence to impair defenders' awareness of malicious activity. Messages produced by defensive tools contain information about potential security events as well as the functioning status of security software and the system. Security reporting messages are important for monitoring the normal operation of a system and identifying important events that can signal a security incident.

T1562.012	Disable or Modify Linux Audit System	Adversaries may disable or modify the Linux audit system to hide malicious activity and avoid detection. Linux admins use the Linux Audit system to track security-relevant information on a system. The Linux Audit system operates at the kernel-level and maintains event logs on application and system activity such as process, network, file, and login events based on pre-configured rules.
T1656	Impersonation	Adversaries may impersonate a trusted person or organization in order to persuade and trick a target into performing some action on their behalf. For example, adversaries may communicate with victims (via Phishing for Information, Phishing, or Internal Spearphishing) while impersonating a known sender such as an executive, colleague, or third-party vendor. Established trust can then be leveraged to accomplish an adversary’s ultimate goals, possibly against multiple victims.
T1070	Indicator Removal	Adversaries may delete or modify artifacts generated within systems to remove evidence of their presence or hinder defenses. Various artifacts may be created by an adversary or something that can be attributed to an adversary’s actions. Typically these artifacts are used as defensive indicators related to monitored events, such as strings from downloaded files, logs that are generated from user actions, and other data analyzed by defenders. Location, format, and type of artifact (such as command or login history) are often specific to each platform.
T1070.001	Clear Windows Event Logs	Adversaries may clear Windows Event Logs to hide the activity of an intrusion. Windows Event Logs are a record of a computer's alerts and notifications. There are three system-defined sources of events: System, Application, and Security, with five event types: Error, Warning, Information, Success Audit, and Failure Audit.

T1070.002	Clear Linux or Mac System Logs	Adversaries may clear system logs to hide evidence of an intrusion. macOS and Linux both keep track of system or user-initiated actions via system logs. The majority of native system logging is stored under the /var/log/ directory. Subfolders in this directory categorize logs by their related functions, such as:
T1070.003	Clear Command History	In addition to clearing system logs, an adversary may clear the command history of a compromised account to conceal the actions undertaken during an intrusion. Various command interpreters keep track of the commands users type in their terminal so that users can retrace what they've done.
T1070.004	File Deletion	Adversaries may delete files left behind by the actions of their intrusion activity. Malware, tools, or other non-native files dropped or created on a system by an adversary (ex: Ingress Tool Transfer) may leave traces to indicate to what was done within a network and how. Removal of these files can occur during an intrusion, or as part of a post-intrusion process to minimize the adversary's footprint.
T1070.005	Network Share Connection Removal	Adversaries may remove share connections that are no longer useful in order to clean up traces of their operation. Windows shared drive and SMB/Windows Admin Shares connections can be removed when no longer needed. Net is an example utility that can be used to remove network share connections with the net use \system\share /delete command.
T1070.006	Timestomp	Adversaries may modify file time attributes to hide new files or changes to existing files. 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 and blend malicious files with legitimate files.

T1070.007	Clear Network Connection History and Configurations	Adversaries may clear or remove evidence of malicious network connections in order to clean up traces of their operations. Configuration settings as well as various artifacts that highlight connection history may be created on a system and/or in application logs from behaviors that require network connections, such as Remote Services or External Remote Services. Defenders may use these artifacts to monitor or otherwise analyze network connections created by adversaries.
T1070.008	Clear Mailbox Data	Adversaries may modify mail and mail application data to remove evidence of their activity. Email applications allow users and other programs to export and delete mailbox data via command line tools or use of APIs. Mail application data can be emails, email metadata, or logs generated by the application or operating system, such as export requests.
T1070.009	Clear Persistence	Adversaries may clear artifacts associated with previously established persistence on a host system to remove evidence of their activity. This may involve various actions, such as removing services, deleting executables, Modify Registry, Plist File Modification, or other methods of cleanup to prevent defenders from collecting evidence of their persistent presence. Adversaries may also delete accounts previously created to maintain persistence (i.e. Create Account).
T1070.01	Relocate Malware	Once a payload is delivered, adversaries may reproduce copies of the same malware on the victim system to remove evidence of their presence and/or avoid defenses. Copying malware payloads to new locations may also be combined with File Deletion to cleanup older artifacts.

T1202	Indirect Command Execution	Adversaries may abuse utilities that allow for command execution to bypass security restrictions that limit the use of command-line interpreters. Various Windows utilities may be used to execute commands, possibly without invoking cmd. For example, Forfiles, the Program Compatibility Assistant (pcalua.exe), components of the Windows Subsystem for Linux (WSL), Scriptrunner.exe, as well as other utilities may invoke the execution of programs and commands from a Command and Scripting Interpreter, Run window, or via scripts.
T1036	Masquerading	Adversaries may attempt to manipulate features of their artifacts to make them appear legitimate or benign to users and/or security tools. Masquerading occurs when the name or location of an object, legitimate or malicious, is manipulated or abused for the sake of evading defenses and observation. This may include manipulating file metadata, tricking users into misidentifying the file type, and giving legitimate task or service names.
T1036.001	Invalid Code Signature	Adversaries may attempt to mimic features of valid code signatures to increase the chance of deceiving a user, analyst, or tool. Code signing provides a level of authenticity on a binary from the developer and a guarantee that the binary has not been tampered with. Adversaries can copy the metadata and signature information from a signed program, then use it as a template for an unsigned program. Files with invalid code signatures will fail digital signature validation checks, but they may appear more legitimate to users and security tools may improperly handle these files.

T1036.002	Right-to-Left Override	Adversaries may abuse the right-to-left override (RTLO or RLO) character (U+202E) to disguise a string and/or file name to make it appear benign. RTLO is a non-printing Unicode character that causes the text that follows it to be displayed in reverse. For example, a Windows screensaver executable named March 25 \u202Excod.scr will display as March 25 rcs.docx. A JavaScript file named photo_high_re\u202Egnp.js will be displayed as photo_high_resj.png.
T1036.003	Rename System Utilities	Adversaries may rename legitimate system utilities to try to evade security mechanisms concerning the usage of those utilities. Security monitoring and control mechanisms may be in place for system utilities adversaries are capable of abusing. It may be possible to bypass those security mechanisms by renaming the utility prior to utilization (ex: rename rundll32.exe). An alternative case occurs when a legitimate utility is copied or moved to a different directory and renamed to avoid detections based on system utilities executing from non-standard paths.
T1036.004	Masquerade Task or Service	Adversaries may attempt to manipulate the name of a task or service to make it appear legitimate or benign. Tasks/services executed by the Task Scheduler or systemd will typically be given a name and/or description. Windows services will have a service name as well as a display name. Many benign tasks and services exist that have commonly associated names. Adversaries may give tasks or services names that are similar or identical to those of legitimate ones.

T1036.005	Match Legitimate Name or Location	Adversaries may match or approximate the name or location of legitimate files or resources when naming/placing them. This is done for the sake of evading defenses and observation. This may be done by placing an executable in a commonly trusted directory (ex: under System32) or giving it the name of a legitimate, trusted program (ex: svchost.exe). In containerized environments, this may also be done by creating a resource in a namespace that matches the naming convention of a container pod or cluster. Alternatively, a file or container image name given may be a close approximation to legitimate programs/images or something innocuous.
T1036.006	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.
T1036.007	Double File Extension	Adversaries may abuse a double extension in the filename as a means of masquerading the true file type. A file name may include a secondary file type extension that may cause only the first extension to be displayed (ex: File.txt.exe may render in some views as just File.txt). However, the second extension is the true file type that determines how the file is opened and executed. The real file extension may be hidden by the operating system in the file browser (ex: explorer.exe), as well as in any software configured using or similar to the system’s policies.

T1036.008	Masquerade File Type	Adversaries may masquerade malicious payloads as legitimate files through changes to the payload's formatting, including the file’s signature, extension, and contents. Various file types have a typical standard format, including how they are encoded and organized. For example, a file’s signature (also known as header or magic bytes) is the beginning bytes of a file and is often used to identify the file’s type. For example, the header of a JPEG file, is 0xFF 0xD8 and the file extension is either .JPE, .JPEG or .JPG.
T1036.009	Break Process Trees	An adversary may attempt to evade process tree-based analysis by modifying executed malware's parent process ID (PPID). If endpoint protection software leverages the "parent-child" relationship for detection, breaking this relationship could result in the adversary’s behavior not being associated with previous process tree activity. On Unix-based systems breaking this process tree is common practice for administrators to execute software using scripts and programs.
T1036.01	Masquerade Account Name	Adversaries may match or approximate the names of legitimate accounts to make newly created ones appear benign. This will typically occur during Create Account, although accounts may also be renamed at a later date. This may also coincide with Account Access Removal if the actor first deletes an account before re-creating one with the same name.
T1556	Modify Authentication Process	Adversaries may modify authentication mechanisms and processes to access user credentials or enable otherwise unwarranted access to accounts. The authentication process is handled by mechanisms, such as the Local Security Authentication Server (LSASS) process and the Security Accounts Manager (SAM) on Windows, pluggable authentication modules (PAM) on Unix-based systems, and authorization plugins on MacOS systems, responsible for gathering, storing, and validating credentials. By modifying an authentication process, an adversary may be able to authenticate to a service or system without using Valid Accounts.

T1556.001	Domain Controller Authentication	Adversaries may patch the authentication process on a domain controller to bypass the typical authentication mechanisms and enable access to accounts.
T1556.002	Password Filter DLL	Adversaries may register malicious password filter dynamic link libraries (DLLs) into the authentication process to acquire user credentials as they are validated.
T1556.003	Pluggable Authentication Modules	Adversaries may modify pluggable authentication modules (PAM) to access user credentials or enable otherwise unwarranted access to accounts. PAM is a modular system of configuration files, libraries, and executable files which guide authentication for many services. The most common authentication module is pam_unix.so, which retrieves, sets, and verifies account authentication information in /etc/passwd and /etc/shadow.
T1556.004	Network Device Authentication	Adversaries may use Patch System Image to hard code a password in the operating system, thus bypassing of native authentication mechanisms for local accounts on network devices.
T1556.005	Reversible Encryption	An adversary may abuse Active Directory authentication encryption properties to gain access to credentials on Windows systems. The AllowReversiblePasswordEncryption property specifies whether reversible password encryption for an account is enabled or disabled. By default this property is disabled (instead storing user credentials as the output of one-way hashing functions) and should not be enabled unless legacy or other software require it.
T1556.006	Multi-Factor Authentication	Adversaries may disable or modify multi-factor authentication (MFA) mechanisms to enable persistent access to compromised accounts.
T1556.007	Hybrid Identity	Adversaries may patch, modify, or otherwise backdoor cloud authentication processes that are tied to on-premises user identities in order to bypass typical authentication mechanisms, access credentials, and enable persistent access to accounts.

T1556.008	Network Provider DLL	Adversaries may register malicious network provider dynamic link libraries (DLLs) to capture cleartext user credentials during the authentication process. Network provider DLLs allow Windows to interface with specific network protocols and can also support add-on credential management functions. During the logon process, Winlogon (the interactive logon module) sends credentials to the local mpnotify.exe process via RPC. The mpnotify.exe process then shares the credentials in cleartext with registered credential managers when notifying that a logon event is happening.
T1556.009	Conditional Access Policies	Adversaries may disable or modify conditional access policies to enable persistent access to compromised accounts. Conditional access policies are additional verifications used by identity providers and identity and access management systems to determine whether a user should be granted access to a resource.
T1578	Modify Cloud Compute Infrastructure	An adversary may attempt to modify a cloud account's compute service infrastructure to evade defenses. A modification to the compute service infrastructure can include the creation, deletion, or modification of one or more components such as compute instances, virtual machines, and snapshots.
T1578.001	Create Snapshot	An adversary may create a snapshot or data backup within a cloud account to evade defenses. A snapshot is a point-in-time copy of an existing cloud compute component such as a virtual machine (VM), virtual hard drive, or volume. An adversary may leverage permissions to create a snapshot in order to bypass restrictions that prevent access to existing compute service infrastructure, unlike in Revert Cloud Instance where an adversary may revert to a snapshot to evade detection and remove evidence of their presence.

T1578.002	Create Cloud Instance	An adversary may create a new instance or virtual machine (VM) within the compute service of a cloud account to evade defenses. Creating a new instance may allow an adversary to bypass firewall rules and permissions that exist on instances currently residing within an account. An adversary may Create Snapshot of one or more volumes in an account, create a new instance, mount the snapshots, and then apply a less restrictive security policy to collect Data from Local System or for Remote Data Staging.
T1578.003	Delete Cloud Instance	An adversary may delete a cloud instance after they have performed malicious activities in an attempt to evade detection and remove evidence of their presence. Deleting an instance or virtual machine can remove valuable forensic artifacts and other evidence of suspicious behavior if the instance is not recoverable.
T1578.004	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 accomplished by restoring virtual machine (VM) or data storage snapshots through the cloud management dashboard or cloud APIs.
T1578.005	Modify Cloud Compute Configurations	Adversaries may modify settings that directly affect the size, locations, and resources available to cloud compute infrastructure in order to evade defenses. These settings may include service quotas, subscription associations, tenant-wide policies, or other configurations that impact available compute. Such modifications may allow adversaries to abuse the victim’s compute resources to achieve their goals, potentially without affecting the execution of running instances and/or revealing their activities to the victim.
T1666	Modify Cloud Resource Hierarchy	Adversaries may attempt to modify hierarchical structures in infrastructure-as-a-service (IaaS) environments in order to evade defenses.

T1112	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.
T1601	Modify System Image	Adversaries may make changes to the operating system of embedded network devices to weaken defenses and provide new capabilities for themselves. On such devices, the operating systems are typically monolithic and most of the device functionality and capabilities are contained within a single file.
T1601.001	Patch System Image	Adversaries may modify the operating system of a network device to introduce new capabilities or weaken existing defenses. Some network devices are built with a monolithic architecture, where the entire operating system and most of the functionality of the device is contained within a single file. Adversaries may change this file in storage, to be loaded in a future boot, or in memory during runtime.
T1601.002	Downgrade System Image	Adversaries may install an older version of the operating system of a network device to weaken security. Older operating system versions on network devices often have weaker encryption ciphers and, in general, fewer/less updated defensive features.
T1599	Network Boundary Bridging	Adversaries may bridge network boundaries by compromising perimeter network devices or internal devices responsible for network segmentation. Breaching these devices may enable an adversary to bypass restrictions on traffic routing that otherwise separate trusted and untrusted networks.
T1599.001	Network Address Translation Traversal	Adversaries may bridge network boundaries by modifying a network device's Network Address Translation (NAT) configuration. Malicious modifications to NAT may enable an adversary to bypass restrictions on traffic routing that otherwise separate trusted and untrusted networks.

T1027	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.
T1027.001	Binary Padding	Adversaries may use binary padding to add junk data and change the on-disk representation of malware. This can be done without affecting the functionality or behavior of a binary, but can increase the size of the binary beyond what some security tools are capable of handling due to file size limitations.
T1027.002	Software Packing	Adversaries may perform software packing or virtual machine software protection to conceal their code. 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. 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.
T1027.003	Steganography	Adversaries may use steganography techniques in order to prevent the detection of hidden information. Steganographic techniques can be used to hide data in digital media such as images, audio tracks, video clips, or text files.
T1027.004	Compile After Delivery	Adversaries may attempt to make payloads difficult to discover and analyze by delivering files to victims as uncompiled code. 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 ilasm.exe, csc.exe, or GCC/MinGW.

T1027.005	Indicator Removal from Tools	Adversaries may remove indicators from tools if they believe their malicious tool was detected, quarantined, or otherwise curtailed. They can modify the tool by removing the indicator and using the updated version that is no longer detected by the target's defensive systems or subsequent targets that may use similar systems.
T1027.006	HTML Smuggling	Adversaries may smuggle data and files past content filters by hiding malicious payloads inside of seemingly benign HTML files. HTML documents can store large binary objects known as JavaScript Blobs (immutable data that represents raw bytes) that can later be constructed into file-like objects. Data may also be stored in Data URLs, which enable embedding media type or MIME files inline of HTML documents. HTML5 also introduced a download attribute that may be used to initiate file downloads.
T1027.007	Dynamic API Resolution	Adversaries may obfuscate then dynamically resolve API functions called by their malware in order to conceal malicious functionalities and impair defensive analysis. Malware commonly uses various Native API functions provided by the OS to perform various tasks such as those involving processes, files, and other system artifacts.
T1027.008	Stripped Payloads	Adversaries may attempt to make a payload difficult to analyze by removing symbols, strings, and other human readable information. Scripts and executables may contain variables names and other strings that help developers document code functionality. Symbols are often created by an operating system’s linker when executable payloads are compiled. Reverse engineers use these symbols and strings to analyze code and to identify functionality in payloads.

T1027.009	Embedded Payloads	Adversaries may embed payloads within other files to conceal malicious content from defenses. Otherwise seemingly benign files (such as scripts and executables) may be abused to carry and obfuscate malicious payloads and content. In some cases, embedded payloads may also enable adversaries to Subvert Trust Controls by not impacting execution controls such as digital signatures and notarization tickets.
T1027.01	Command Obfuscation	Adversaries may obfuscate content during command execution to impede detection. Command-line obfuscation is a method of making strings and patterns within commands and scripts more difficult to signature and analyze. This type of obfuscation can be included within commands executed by delivered payloads (e.g., Phishing and Drive-by Compromise) or interactively via Command and Scripting Interpreter.
T1027.011	Fileless Storage	Adversaries may store data in "fileless" formats to conceal malicious activity from defenses. Fileless storage can be broadly defined as any format other than a file. Common examples of non-volatile fileless storage in Windows systems include the Windows Registry, event logs, or WMI repository. In Linux systems, shared memory directories such as /dev/shm, /run/shm, /var/run, and /var/lock may also be considered fileless storage, as files written to these directories are mapped directly to RAM and not stored on the disk.
T1027.012	LNK Icon Smuggling	Adversaries may smuggle commands to download malicious payloads past content filters by hiding them within otherwise seemingly benign windows shortcut files. Windows shortcut files (.LNK) include many metadata fields, including an icon location field (also known as the IconEnvironmentDataBlock) designed to specify the path to an icon file that is to be displayed for the LNK file within a host directory.

T1027.013	Encrypted/Encoded File	Adversaries may encrypt or encode files to obfuscate strings, bytes, and other specific patterns to impede detection. Encrypting and/or encoding file content aims to conceal malicious artifacts within a file used in an intrusion. Many other techniques, such as Software Packing, Steganography, and Embedded Payloads, share this same broad objective. Encrypting and/or encoding files could lead to a lapse in detection of static signatures, only for this malicious content to be revealed (i.e., Deobfuscate/Decode Files or Information) at the time of execution/use.
T1027.014	Polymorphic Code	Adversaries may utilize polymorphic code (also known as metamorphic or mutating code) to evade detection. Polymorphic code is a type of software capable of changing its runtime footprint during code execution. With each execution of the software, the code is mutated into a different version of itself that achieves the same purpose or objective as the original. This functionality enables the malware to evade traditional signature-based defenses, such as antivirus and antimalware tools. Other obfuscation techniques can be used in conjunction with polymorphic code to accomplish the intended effects, including using mutation engines to conduct actions such as Software Packing, Command Obfuscation, or Encrypted/Encoded File.
T1647	Plist File Modification	Adversaries may modify property list files (plist files) to enable other malicious activity, while also potentially evading and bypassing system defenses. macOS applications use plist files, such as the info.plist file, to store properties and configuration settings that inform the operating system how to handle the application at runtime. Plist files are structured metadata in key-value pairs formatted in XML based on Apple's Core Foundation DTD. Plist files can be saved in text or binary format.

T1542	Pre-OS Boot	Adversaries may abuse Pre-OS Boot mechanisms as a way to establish persistence on a system. During the booting process of a computer, firmware and various startup services are loaded before the operating system. These programs control flow of execution before the operating system takes control.
T1542.001	System Firmware	Adversaries may modify system firmware to persist on systems.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.
T1542.002	Component Firmware	Adversaries may modify component firmware to persist on systems. 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 but conducted upon other system components/devices that may not have the same capability or level of integrity checking.
T1542.003	Bootkit	Adversaries may use bootkits to persist on systems. Bootkits reside at a layer below the operating system and may make it difficult to perform full remediation unless an organization suspects one was used and can act accordingly.
T1542.004	ROMMONkit	Adversaries may abuse the ROM Monitor (ROMMON) by loading an unauthorized firmware with adversary code to provide persistent access and manipulate device behavior that is difficult to detect.
T1542.005	TFTP Boot	Adversaries may abuse netbooting to load an unauthorized network device operating system from a Trivial File Transfer Protocol (TFTP) server. TFTP boot (netbooting) is commonly used by network administrators to load configuration-controlled network device images from a centralized management server. Netbooting is one option in the boot sequence and can be used to centralize, manage, and control device images.

T1055	Process Injection	Adversaries may inject code into processes in order to evade process-based defenses as well as possibly elevate privileges. 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.
T1055.001	Dynamic-link Library Injection	Adversaries may inject dynamic-link libraries (DLLs) into processes in order to evade process-based defenses as well as possibly elevate privileges. DLL injection is a method of executing arbitrary code in the address space of a separate live process.
T1055.002	Portable Executable Injection	Adversaries may inject portable executables (PE) into processes in order to evade process-based defenses as well as possibly elevate privileges. PE injection is a method of executing arbitrary code in the address space of a separate live process.
T1055.003	Thread Execution Hijacking	Adversaries may inject malicious code into hijacked processes in order to evade process-based defenses as well as possibly elevate privileges. Thread Execution Hijacking is a method of executing arbitrary code in the address space of a separate live process.
T1055.004	Asynchronous Procedure Call	Adversaries may inject malicious code into processes via the asynchronous procedure call (APC) queue in order to evade process-based defenses as well as possibly elevate privileges. APC injection is a method of executing arbitrary code in the address space of a separate live process.
T1055.005	Thread Local Storage	Adversaries may inject malicious code into processes via thread local storage (TLS) callbacks in order to evade process-based defenses as well as possibly elevate privileges. TLS callback injection is a method of executing arbitrary code in the address space of a separate live process.

T1055.008	Ptrace System Calls	Adversaries may inject malicious code into processes via ptrace (process trace) system calls in order to evade process-based defenses as well as possibly elevate privileges. Ptrace system call injection is a method of executing arbitrary code in the address space of a separate live process.
T1055.009	Proc Memory	Adversaries may inject malicious code into processes via the /proc filesystem in order to evade process-based defenses as well as possibly elevate privileges. Proc memory injection is a method of executing arbitrary code in the address space of a separate live process.
T1055.011	Extra Window Memory Injection	Adversaries may inject malicious code into process via Extra Window Memory (EWM) in order to evade process-based defenses as well as possibly elevate privileges. EWM injection is a method of executing arbitrary code in the address space of a separate live process.
T1055.012	Process Hollowing	Adversaries may inject malicious code into suspended and hollowed processes in order to evade process-based defenses. Process hollowing is a method of executing arbitrary code in the address space of a separate live process.
T1055.013	Process Doppelgänging	Adversaries may inject malicious code into process via process doppelgänging in order to evade process-based defenses as well as possibly elevate privileges. Process doppelgänging is a method of executing arbitrary code in the address space of a separate live process.
T1055.014	VDSO Hijacking	Adversaries may inject malicious code into processes via VDSO hijacking in order to evade process-based defenses as well as possibly elevate privileges. Virtual dynamic shared object (vdso) hijacking is a method of executing arbitrary code in the address space of a separate live process.

T1055.015	ListPlanting	Adversaries may abuse list-view controls to inject malicious code into hijacked processes in order to evade process-based defenses as well as possibly elevate privileges. ListPlanting is a method of executing arbitrary code in the address space of a separate live process. Code executed via ListPlanting may also evade detection from security products since the execution is masked under a legitimate process.
T1620	Reflective Code Loading	Adversaries may reflectively load code into a process in order to conceal the execution of malicious payloads. Reflective loading involves allocating then executing payloads directly within the memory of the process, vice creating a thread or process backed by a file path on disk (e.g., Shared Modules).
T1207	Rogue Domain Controller	Adversaries may register a rogue Domain Controller to enable manipulation of Active Directory data. DCShadow may be used to create a rogue Domain Controller (DC). 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 DC. Once registered, a rogue DC may be able to inject and replicate changes into AD infrastructure for any domain object, including credentials and keys.
T1014	Rootkit	Adversaries may use rootkits to hide the presence of programs, files, network connections, services, drivers, and other system components. Rootkits are programs that hide the existence of malware by intercepting/hooking and modifying operating system API calls that supply system information.

T1553	Subvert Trust Controls	Adversaries may undermine security controls that will either warn users of untrusted activity or prevent execution of untrusted programs. Operating systems and security products may contain mechanisms to identify programs or websites as possessing some level of trust. Examples of such features would include a program being allowed to run because it is signed by a valid code signing certificate, a program prompting the user with a warning because it has an attribute set from being downloaded from the Internet, or getting an indication that you are about to connect to an untrusted site.
T1553.001	Gatekeeper Bypass	Adversaries may modify file attributes and subvert Gatekeeper functionality to evade user prompts and execute untrusted programs. Gatekeeper is a set of technologies that act as layer of Apple’s security model to ensure only trusted applications are executed on a host. Gatekeeper was built on top of File Quarantine in Snow Leopard (10.6, 2009) and has grown to include Code Signing, security policy compliance, Notarization, and more. Gatekeeper also treats applications running for the first time differently than reopened applications.
T1553.002	Code Signing	Adversaries may create, acquire, or steal code signing materials to sign their malware or tools. Code signing provides a level of authenticity on a binary from the developer and a guarantee that the binary has not been tampered with. The certificates used during an operation may be created, acquired, or stolen by the adversary. Unlike Invalid Code Signature, this activity will result in a valid signature.

T1553.003	SIP and Trust Provider Hijacking	Adversaries may tamper with SIP and trust provider components to mislead the operating system and application control tools when conducting signature validation checks. In user mode, Windows 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, which accepts an inquiry and coordinates with the appropriate trust provider, which is responsible for validating parameters of a signature.
T1553.004	Install Root Certificate	Adversaries may install a root certificate on a compromised system to avoid warnings when connecting to adversary controlled web servers. 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. 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.

T1553.005	Mark-of-the-Web Bypass	Adversaries may abuse specific file formats to subvert Mark-of-the-Web (MOTW) controls. In Windows, when files are downloaded from the Internet, they are tagged with a hidden NTFS Alternate Data Stream (ADS) named Zone.Identifier with a specific value known as the MOTW. Files that are tagged with MOTW are protected and cannot perform certain actions. For example, starting in MS Office 10, if a MS Office file has the MOTW, it will open in Protected View. Executables tagged with the MOTW will be processed by Windows Defender SmartScreen that compares files with an allowlist of well-known executables. If the file is not known/trusted, SmartScreen will prevent the execution and warn the user not to run it.
T1553.006	Code Signing Policy Modification	Adversaries may modify code signing policies to enable execution of unsigned or self-signed code. Code signing provides a level of authenticity on a program from a developer and a guarantee that the program has not been tampered with. Security controls can include enforcement mechanisms to ensure that only valid, signed code can be run on an operating system.
T1218	System Binary Proxy Execution	Adversaries may bypass process and/or signature-based defenses by proxying execution of malicious content with signed, or otherwise trusted, binaries. Binaries used in this technique are often Microsoft-signed files, indicating that they have been either downloaded from Microsoft or are already native in the operating system. Binaries signed with trusted digital certificates can typically 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 or commands.

T1218.001	Compiled HTML File	Adversaries may abuse Compiled HTML files (.chm) to conceal malicious code. CHM files 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. CHM content is displayed using underlying components of the Internet Explorer browser loaded by the HTML Help executable program (hh.exe).
T1218.002	Control Panel	Adversaries may abuse control.exe to proxy execution of malicious payloads. The Windows Control Panel process binary (control.exe) handles execution of Control Panel items, which are utilities that allow users to view and adjust computer settings.
T1218.003	CMSTP	Adversaries may abuse CMSTP to proxy execution of malicious code. The Microsoft Connection Manager Profile Installer (CMSTP.exe) is a command-line program used to install Connection Manager service profiles. CMSTP.exe accepts an installation information file (INF) as a parameter and installs a service profile leveraged for remote access connections.
T1218.004	InstallUtil	Adversaries may use InstallUtil to proxy execution of code through a trusted Windows utility. InstallUtil is a command-line utility that allows for installation and uninstallation of resources by executing specific installer components specified in .NET binaries. The InstallUtil binary may also be digitally signed by Microsoft and located in the .NET directories on a Windows system: C:\Windows\Microsoft.NET\Framework\v\InstallUtil.exe and C:\Windows\Microsoft.NET\Framework64\v\InstallUtil.exe.
T1218.005	Mshta	Adversaries may abuse 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

T1218.007	Msiexec	Adversaries may abuse msiexec.exe to proxy execution of malicious payloads. Msiexec.exe is the command-line utility for the Windows Installer and is thus commonly associated with executing installation packages (.msi). The Msiexec.exe binary may also be digitally signed by Microsoft.
T1218.008	Odbcconf	Adversaries may abuse odbcconf.exe to proxy execution of malicious payloads. Odbcconf.exe is a Windows utility that allows you to configure Open Database Connectivity (ODBC) drivers and data source names. The Odbcconf.exe binary may be digitally signed by Microsoft.
T1218.009	Regsvcs/Regasm	Adversaries may abuse Regsvcs and Regasm to proxy execution of code through a trusted Windows utility. Regsvcs and Regasm are Windows command-line utilities that are used to register .NET Component Object Model (COM) assemblies. Both are binaries that may be digitally signed by Microsoft.
T1218.01	Regsvr32	Adversaries may abuse Regsvr32.exe to proxy execution of malicious code. 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. The Regsvr32.exe binary may also be signed by Microsoft.
T1218.011	Rundll32	Adversaries may abuse rundll32.exe to proxy execution of malicious code. Using rundll32.exe, vice executing directly (i.e. Shared Modules), may avoid triggering security tools that may not monitor execution of the rundll32.exe process because of allowlists or false positives from normal operations. Rundll32.exe is commonly associated with executing DLL payloads (ex: rundll32.exe {DLLname, DLLfunction}).
T1218.012	Verclsid	Adversaries may abuse verclsid.exe to proxy execution of malicious code. Verclsid.exe is known as the Extension CLSID Verification Host and is responsible for verifying each shell extension before they are used by Windows Explorer or the Windows Shell.

T1218.013	Mavinject	Adversaries may abuse mavinject.exe to proxy execution of malicious code. Mavinject.exe is the Microsoft Application Virtualization Injector, a Windows utility that can inject code into external processes as part of Microsoft Application Virtualization (App-V).
T1218.014	MMC	Adversaries may abuse mmc.exe to proxy execution of malicious .msc files. Microsoft Management Console (MMC) is a binary that may be signed by Microsoft and is used in several ways in either its GUI or in a command prompt. MMC can be used to create, open, and save custom consoles that contain administrative tools created by Microsoft, called snap-ins. These snap-ins may be used to manage Windows systems locally or remotely. MMC can also be used to open Microsoft created .msc files to manage system configuration.
T1218.015	Electron Applications	Adversaries may abuse components of the Electron framework to execute malicious code. The Electron framework hosts many common applications such as Signal, Slack, and Microsoft Teams. Originally developed by GitHub, Electron is a cross-platform desktop application development framework that employs web technologies like JavaScript, HTML, and CSS. The Chromium engine is used to display web content and Node.js runs the backend code.
T1216	System Script Proxy Execution	Adversaries may use trusted scripts, often signed with certificates, to proxy the execution of malicious files. Several Microsoft signed scripts that have been downloaded from Microsoft or 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 control and signature validation on systems.

T1216.001	PubPrn	Adversaries may use PubPrn to proxy execution of malicious remote files. PubPrn.vbs is a Visual Basic script that publishes a printer to Active Directory Domain Services. The script may be signed by Microsoft and is commonly executed through the Windows Command Shell via Cscript.exe. For example, the following code publishes a printer within the specified domain: cscript pubprn Printer1 LDAP://CN=Container1,DC=Domain1,DC=Com.
T1216.002	SyncAppvPublishingServer	Adversaries may abuse SyncAppvPublishingServer.vbs to proxy execution of malicious PowerShell commands. SyncAppvPublishingServer.vbs is a Visual Basic script associated with how Windows virtualizes applications (Microsoft Application Virtualization, or App-V). For example, Windows may render Win32 applications to users as virtual applications, allowing users to launch and interact with them as if they were installed locally.
T1221	Template Injection	Adversaries may create or modify references in user document templates to conceal malicious code or force authentication attempts. For example, Microsoft's Office Open 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.

T1205	Traffic Signaling	Adversaries may use traffic signaling to hide open ports or other malicious functionality used for persistence or command and control. Traffic signaling involves the use of a magic value or sequence that must be sent to a system to trigger a special response, such as opening a closed port or executing a malicious task. This may take the form of sending a series of packets with certain characteristics before a port will be opened that the adversary can use for command and control. Usually this series of packets consists of attempted connections to a predefined sequence of closed ports (i.e. Port Knocking), but can involve unusual flags, specific strings, or other unique characteristics. After the sequence is completed, opening a port may be accomplished by the host-based firewall, but could also be implemented by custom software.
T1205.001	Port Knocking	Adversaries may use port knocking to hide open ports used for persistence or command and control. To enable a port, an adversary sends a series of attempted connections to a predefined sequence of closed ports. After the sequence is completed, opening a port is often accomplished by the host based firewall, but could also be implemented by custom software.
T1205.002	Socket Filters	Adversaries may attach filters to a network socket to monitor then activate backdoors used for persistence or command and control. With elevated permissions, adversaries can use features such as the libpcap library to open sockets and install filters to allow or disallow certain types of data to come through the socket. The filter may apply to all traffic passing through the specified network interface (or every interface if not specified). When the network interface receives a packet matching the filter criteria, additional actions can be triggered on the host, such as activation of a reverse shell.

T1127	Trusted Developer Utilities Proxy Execution	Adversaries may take advantage of trusted developer utilities to proxy execution of malicious payloads. 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 control solutions.
T1127.001	MSBuild	Adversaries may use MSBuild to proxy execution of code through a trusted Windows utility. MSBuild.exe (Microsoft Build Engine) is a software build platform used by Visual Studio. It handles XML formatted project files that define requirements for loading and building various platforms and configurations.
T1127.002	ClickOnce	Adversaries may use ClickOnce applications (.appref-ms and .application files) to proxy execution of code through a trusted Windows utility. ClickOnce is a deployment that enables a user to create self-updating Windows-based .NET applications (i.e, .XBAP, .EXE, or .DLL) that install and run from a file share or web page with minimal user interaction. The application launches as a child process of DFSVC.EXE, which is responsible for installing, launching, and updating the application.
T1535	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.
T1550	Use Alternate Authentication Material	Adversaries may use alternate authentication material, such as password hashes, Kerberos tickets, and application access tokens, in order to move laterally within an environment and bypass normal system access controls.

T1550.001	Application Access Token	Adversaries may use stolen 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 or services and used in lieu of login credentials.
T1550.002	Pass the Hash	Adversaries may "pass the hash" using stolen password hashes to move laterally within an environment, bypassing normal system access controls. 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.
T1550.003	Pass the Ticket	Adversaries may "pass the ticket" using stolen Kerberos tickets to move laterally within an environment, bypassing normal system access controls. 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.
T1550.004	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.

T1078	Valid Accounts	Adversaries may obtain and abuse credentials of existing accounts as a means of gaining Initial Access, Persistence, Privilege Escalation, or Defense Evasion. Compromised credentials may be used to bypass access controls placed on various resources on systems within the network and may even be used for persistent access to remote systems and externally available services, such as VPNs, Outlook Web Access, network devices, 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.
T1078.001	Default Accounts	Adversaries may obtain and abuse credentials of a default account as a means of gaining Initial Access, Persistence, Privilege Escalation, or Defense Evasion. Default accounts are those that are built-into an OS, such as the Guest or Administrator accounts on Windows systems. Default accounts also include default factory/provider set accounts on other types of systems, software, or devices, including the root user account in AWS and the default service account in Kubernetes.
T1078.002	Domain Accounts	Adversaries may obtain and abuse credentials of a domain account as a means of gaining Initial Access, Persistence, Privilege Escalation, or Defense Evasion. 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.
T1078.003	Local Accounts	Adversaries may obtain and abuse credentials of a local account as a means of gaining Initial Access, Persistence, Privilege Escalation, or Defense Evasion. Local accounts are those configured by an organization for use by users, remote support, services, or for administration on a single system or service.

T1078.004	Cloud Accounts	Valid accounts in cloud environments may allow adversaries to perform actions to achieve Initial Access, Persistence, Privilege Escalation, or Defense Evasion. Cloud accounts are those created and configured by an organization for use by users, remote support, services, or for administration of resources within a cloud service provider or SaaS application. Cloud Accounts can exist solely in the cloud; alternatively, they may be hybrid-joined between on-premises systems and the cloud through syncing or federation with other identity sources such as Windows Active Directory.
T1497	Virtualization/Sandbox Evasion	Adversaries may employ various means to detect and avoid virtualization and analysis environments. This may include changing behaviors based on the results of checks for the presence of artifacts indicative of a virtual machine environment (VME) or sandbox. If the adversary detects a VME, they may alter their malware to disengage from the victim or conceal the core functions of the implant. They may also search for VME artifacts before dropping secondary or additional payloads. Adversaries may use the information learned from Virtualization/Sandbox Evasion during automated discovery to shape follow-on behaviors.
T1497.001	System Checks	Adversaries may employ various system checks to detect and avoid virtualization and analysis environments. This may include changing behaviors based on the results of checks for the presence of artifacts indicative of a virtual machine environment (VME) or sandbox. If the adversary detects a VME, they may alter their malware to disengage from the victim or conceal the core functions of the implant. They may also search for VME artifacts before dropping secondary or additional payloads. Adversaries may use the information learned from Virtualization/Sandbox Evasion during automated discovery to shape follow-on behaviors.

T1497.002	User Activity Based Checks	Adversaries may employ various user activity checks to detect and avoid virtualization and analysis environments. This may include changing behaviors based on the results of checks for the presence of artifacts indicative of a virtual machine environment (VME) or sandbox. If the adversary detects a VME, they may alter their malware to disengage from the victim or conceal the core functions of the implant. They may also search for VME artifacts before dropping secondary or additional payloads. Adversaries may use the information learned from Virtualization/Sandbox Evasion during automated discovery to shape follow-on behaviors.
T1497.003	Time Based Evasion	Adversaries may employ various time-based methods to detect and avoid virtualization and analysis environments. This may include enumerating time-based properties, such as uptime or the system clock, as well as the use of timers or other triggers to avoid a virtual machine environment (VME) or sandbox, specifically those that are automated or only operate for a limited amount of time.
T1600	Weaken Encryption	Adversaries may compromise a network device's encryption capability in order to bypass encryption that would otherwise protect data communications.
T1600.001	Reduce Key Space	Adversaries may reduce the level of effort required to decrypt data transmitted over the network by reducing the cipher strength of encrypted communications.
T1600.002	Disable Crypto Hardware	Adversaries disable a network device's dedicated hardware encryption, which may enable them to leverage weaknesses in software encryption in order to reduce the effort involved in collecting, manipulating, and exfiltrating transmitted data.

T1220	XSL Script Processing	Adversaries may bypass application control and obscure execution of code by embedding scripts inside XSL files. 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.

Tactic: Credential Access
ID: TA0006
Description:
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.

Techniques Under the Credential Access Tactic

ID	Name	Description
T1557	Adversary-in-the-Middle	Adversaries may attempt to position themselves between two or more networked devices using an adversary-in-the-middle (AiTM) technique to support follow-on behaviors such as Network Sniffing, Transmitted Data Manipulation, or replay attacks (Exploitation for Credential Access). By abusing features of common networking protocols that can determine the flow of network traffic (e.g. ARP, DNS, LLMNR, etc.), adversaries may force a device to communicate through an adversary controlled system so they can collect information or perform additional actions.
T1557.001	LLMNR/NBT-NS Poisoning and SMB Relay	By responding to LLMNR/NBT-NS network traffic, adversaries may spoof an authoritative source for name resolution to force communication with an adversary controlled system. This activity may be used to collect or relay authentication materials.
T1557.002	ARP Cache Poisoning	Adversaries may poison Address Resolution Protocol (ARP) caches to position themselves between the communication of two or more networked devices. This activity may be used to enable follow-on behaviors such as Network Sniffing or Transmitted Data Manipulation.
T1557.003	DHCP Spoofing	Adversaries may redirect network traffic to adversary-owned systems by spoofing Dynamic Host Configuration Protocol (DHCP) traffic and acting as a malicious DHCP server on the victim network. By achieving the adversary-in-the-middle (AiTM) position, adversaries may collect network communications, including passed credentials, especially those sent over insecure, unencrypted protocols. This may also enable follow-on behaviors such as Network Sniffing or Transmitted Data Manipulation.
T1557.004	Evil Twin	Adversaries may host seemingly genuine Wi-Fi access points to deceive users into connecting to malicious networks as a way of supporting follow-on behaviors such as Network Sniffing, Transmitted Data Manipulation, or Input Capture.

T1110	Brute Force	Adversaries may use brute force techniques to gain access to accounts when passwords are unknown or when password hashes are obtained. Without knowledge of the password for an account or set of accounts, an adversary may systematically guess the password using a repetitive or iterative mechanism. Brute forcing passwords can take place via interaction with a service that will check the validity of those credentials or offline against previously acquired credential data, such as password hashes.
T1110.001	Password Guessing	Adversaries with no prior knowledge of legitimate credentials within the system or environment may guess passwords to attempt access to accounts. Without knowledge of the password for an account, an adversary may opt to systematically guess the password using a repetitive or iterative mechanism. An adversary may guess login credentials without prior knowledge of system or environment passwords during an operation by using a list of common passwords. Password guessing may or may not take into account the target's policies on password complexity or use policies that may lock accounts out after a number of failed attempts.
T1110.002	Password Cracking	Adversaries may use password cracking to attempt to recover usable credentials, such as plaintext passwords, when credential material such as password hashes are obtained. OS Credential Dumping can be used to obtain password hashes, this may only get an adversary so far when Pass the Hash is not an option. Further, adversaries may leverage Data from Configuration Repository in order to obtain hashed credentials for network devices.

T1110.003	Password Spraying	Adversaries may use a single or small list of commonly used passwords against many different accounts to attempt to acquire valid account credentials. Password spraying uses one password (e.g. 'Password01'), or a small list of commonly used passwords, that may match the complexity policy of the domain. Logins are attempted with that password against many different accounts on a network to avoid account lockouts that would normally occur when brute forcing a single account with many passwords.
T1110.004	Credential Stuffing	Adversaries may use credentials obtained from breach dumps of unrelated accounts to gain access to target accounts through credential overlap. Occasionally, large numbers of username and password pairs are dumped online when a website or service is compromised and the user account credentials accessed. The information may be useful to an adversary attempting to compromise accounts by taking advantage of the tendency for users to use the same passwords across personal and business accounts.
T1555	Credentials from Password Stores	Adversaries may search for common password storage locations to obtain user credentials. Passwords are stored in several places on a system, depending on the operating system or application holding the credentials. There are also specific applications and services that store passwords to make them easier for users to manage and maintain, such as password managers and cloud secrets vaults. Once credentials are obtained, they can be used to perform lateral movement and access restricted information.

T1555.001	Keychain	Adversaries may acquire credentials from Keychain. Keychain (or Keychain Services) is the macOS credential management system that stores account names, passwords, private keys, certificates, sensitive application data, payment data, and secure notes. There are three types of Keychains: Login Keychain, System Keychain, and Local Items (iCloud) Keychain. The default Keychain is the Login Keychain, which stores user passwords and information. The System Keychain stores items accessed by the operating system, such as items shared among users on a host. The Local Items (iCloud) Keychain is used for items synced with Apple's iCloud service.
T1555.002	Securityd Memory	An adversary with root access may gather credentials by reading securityd's memory. securityd is a service/daemon responsible for implementing security protocols such as encryption and authorization. A privileged adversary may be able to scan through securityd's memory to find the correct sequence of keys to decrypt the user's logon keychain. This may provide the adversary with various plaintext passwords, such as those for users, WiFi, mail, browsers, certificates, secure notes, etc.
T1555.003	Credentials from Web Browsers	Adversaries may acquire credentials from web browsers by reading files specific to the target browser. 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.
T1555.004	Windows Credential Manager	Adversaries may acquire credentials from the Windows Credential Manager. The Credential Manager stores credentials for signing into websites, applications, and/or devices that request authentication through NTLM or Kerberos in Credential Lockers (previously known as Windows Vaults).

T1555.005	Password Managers	Adversaries may acquire user credentials from third-party password managers. Password managers are applications designed to store user credentials, normally in an encrypted database. Credentials are typically accessible after a user provides a master password that unlocks the database. After the database is unlocked, these credentials may be copied to memory. These databases can be stored as files on disk.
T1555.006	Cloud Secrets Management Stores	Adversaries may acquire credentials from cloud-native secret management solutions such as AWS Secrets Manager, GCP Secret Manager, Azure Key Vault, and Terraform Vault.
T1212	Exploitation for Credential Access	Adversaries may exploit software vulnerabilities in an attempt to collect credentials. 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.
T1187	Forced Authentication	Adversaries may gather credential material by invoking or forcing a user to automatically provide authentication information through a mechanism in which they can intercept.
T1606	Forge Web Credentials	Adversaries may forge credential materials that can be used to gain access to web applications or Internet services. Web applications and services (hosted in cloud SaaS environments or on-premise servers) often use session cookies, tokens, or other materials to authenticate and authorize user access.
T1606.001	Web Cookies	Adversaries may forge web cookies that can be used to gain access to web applications or Internet services. Web applications and services (hosted in cloud SaaS environments or on-premise servers) often use session cookies to authenticate and authorize user access.

T1606.002	SAML Tokens	An adversary may forge SAML tokens with any permissions claims and lifetimes if they possess a valid SAML token-signing certificate. The default lifetime of a SAML token is one hour, but the validity period can be specified in the NotOnOrAfter value of the conditions ... element in a token. This value can be changed using the AccessTokenLifetime in a LifetimeTokenPolicy. Forged SAML tokens enable adversaries to authenticate across services that use SAML 2.0 as an SSO (single sign-on) mechanism.
T1056	Input Capture	Adversaries may use methods of capturing user input to obtain credentials or collect information. During normal system usage, users often provide credentials to various different locations, such as login pages/portals or system dialog boxes. Input capture mechanisms may be transparent to the user (e.g. Credential API Hooking) or rely on deceiving the user into providing input into what they believe to be a genuine service (e.g. Web Portal Capture).
T1056.001	Keylogging	Adversaries may log user keystrokes to intercept credentials as the user types them. Keylogging is likely to be used to acquire credentials for new access opportunities when OS Credential Dumping efforts are not effective, and may require an adversary to intercept keystrokes on a system for a substantial period of time before credentials can be successfully captured. In order to increase the likelihood of capturing credentials quickly, an adversary may also perform actions such as clearing browser cookies to force users to reauthenticate to systems.
T1056.002	GUI Input Capture	Adversaries may mimic common operating system GUI components to prompt users for credentials with a seemingly legitimate 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).

T1056.003	Web Portal Capture	Adversaries may 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. For example, a compromised login page may log provided user credentials before logging the user in to the service.
T1056.004	Credential API Hooking	Adversaries may hook into Windows application programming interface (API) functions to collect user credentials. Malicious hooking mechanisms may capture API calls that include parameters that reveal user authentication credentials. Unlike Keylogging, this technique focuses specifically on API functions that include parameters that reveal user credentials. Hooking involves redirecting calls to these functions and can be implemented via:
T1556	Modify Authentication Process	Adversaries may modify authentication mechanisms and processes to access user credentials or enable otherwise unwarranted access to accounts. The authentication process is handled by mechanisms, such as the Local Security Authentication Server (LSASS) process and the Security Accounts Manager (SAM) on Windows, pluggable authentication modules (PAM) on Unix-based systems, and authorization plugins on MacOS systems, responsible for gathering, storing, and validating credentials. By modifying an authentication process, an adversary may be able to authenticate to a service or system without using Valid Accounts.
T1556.001	Domain Controller Authentication	Adversaries may patch the authentication process on a domain controller to bypass the typical authentication mechanisms and enable access to accounts.
T1556.002	Password Filter DLL	Adversaries may register malicious password filter dynamic link libraries (DLLs) into the authentication process to acquire user credentials as they are validated.

T1556.003	Pluggable Authentication Modules	Adversaries may modify pluggable authentication modules (PAM) to access user credentials or enable otherwise unwarranted access to accounts. PAM is a modular system of configuration files, libraries, and executable files which guide authentication for many services. The most common authentication module is pam_unix.so, which retrieves, sets, and verifies account authentication information in /etc/passwd and /etc/shadow.
T1556.004	Network Device Authentication	Adversaries may use Patch System Image to hard code a password in the operating system, thus bypassing of native authentication mechanisms for local accounts on network devices.
T1556.005	Reversible Encryption	An adversary may abuse Active Directory authentication encryption properties to gain access to credentials on Windows systems. The AllowReversiblePasswordEncryption property specifies whether reversible password encryption for an account is enabled or disabled. By default this property is disabled (instead storing user credentials as the output of one-way hashing functions) and should not be enabled unless legacy or other software require it.
T1556.006	Multi-Factor Authentication	Adversaries may disable or modify multi-factor authentication (MFA) mechanisms to enable persistent access to compromised accounts.
T1556.007	Hybrid Identity	Adversaries may patch, modify, or otherwise backdoor cloud authentication processes that are tied to on-premises user identities in order to bypass typical authentication mechanisms, access credentials, and enable persistent access to accounts.

T1556.008	Network Provider DLL	Adversaries may register malicious network provider dynamic link libraries (DLLs) to capture cleartext user credentials during the authentication process. Network provider DLLs allow Windows to interface with specific network protocols and can also support add-on credential management functions. During the logon process, Winlogon (the interactive logon module) sends credentials to the local mpnotify.exe process via RPC. The mpnotify.exe process then shares the credentials in cleartext with registered credential managers when notifying that a logon event is happening.
T1556.009	Conditional Access Policies	Adversaries may disable or modify conditional access policies to enable persistent access to compromised accounts. Conditional access policies are additional verifications used by identity providers and identity and access management systems to determine whether a user should be granted access to a resource.
T1111	Multi-Factor Authentication Interception	Adversaries may target multi-factor authentication (MFA) mechanisms, (i.e., smart cards, token generators, etc.) to gain access to credentials that can be used to access systems, services, and network resources. Use of MFA is recommended and provides a higher level of security than usernames and passwords alone, but organizations should be aware of techniques that could be used to intercept and bypass these security mechanisms.
T1621	Multi-Factor Authentication Request Generation	Adversaries may attempt to bypass multi-factor authentication (MFA) mechanisms and gain access to accounts by generating MFA requests sent to users.
T1040	Network Sniffing	Adversaries may passively sniff network traffic to capture information about an environment, including authentication material passed over the network. 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.

T1003	OS Credential Dumping	Adversaries may attempt to dump credentials to obtain account login and credential material, normally in the form of a hash or a clear text password. Credentials can be obtained from OS caches, memory, or structures. Credentials can then be used to perform Lateral Movement and access restricted information.
T1003.001	LSASS Memory	Adversaries may attempt to access credential material stored in the process memory of the Local Security Authority Subsystem Service (LSASS). After a user logs on, the system generates and stores a variety of credential materials in LSASS process memory. These credential materials can be harvested by an administrative user or SYSTEM and used to conduct Lateral Movement using Use Alternate Authentication Material.
T1003.002	Security Account Manager	Adversaries may attempt to extract credential material from the Security Account Manager (SAM) database either through in-memory techniques or through the Windows Registry where the SAM database is stored. The SAM is a database file that contains local accounts for the host, typically those found with the net user command. Enumerating the SAM database requires SYSTEM level access.
T1003.003	NTDS	Adversaries may attempt to access or create a copy of the Active Directory domain database in order to steal credential information, as well as obtain other information about domain members such as devices, users, and access rights. By default, the NTDS file (NTDS.dit) is located in %SystemRoot%\NTDS\Ntds.dit of a domain controller.
T1003.004	LSA Secrets	Adversaries with SYSTEM access to a host may attempt to access Local Security Authority (LSA) secrets, which can contain a variety of different credential materials, such as credentials for service accounts. LSA secrets are stored in the registry at HKEY_LOCAL_MACHINE\SECURITY\Policy\Secrets. LSA secrets can also be dumped from memory.

T1003.005	Cached Domain Credentials	Adversaries may attempt to access cached domain credentials used to allow authentication to occur in the event a domain controller is unavailable.
T1003.006	DCSync	Adversaries may attempt to access credentials and other sensitive information by abusing a Windows Domain Controller's application programming interface (API) to simulate the replication process from a remote domain controller using a technique called DCSync.
T1003.007	Proc Filesystem	Adversaries may gather credentials from the proc filesystem or /proc. The proc filesystem is a pseudo-filesystem used as an interface to kernel data structures for Linux based systems managing virtual memory. For each process, the /proc/<PID>/maps file shows how memory is mapped within the process's virtual address space. And /proc/<PID>/mem, exposed for debugging purposes, provides access to the process's virtual address space.
T1003.008	/etc/passwd and /etc/shadow	Adversaries may attempt to dump the contents of /etc/passwd and /etc/shadow to enable offline password cracking. Most modern Linux operating systems use a combination of /etc/passwd and /etc/shadow to store user account information including password hashes in /etc/shadow. By default, /etc/shadow is only readable by the root user.
T1528	Steal Application Access Token	Adversaries can steal application access tokens as a means of acquiring credentials to access remote systems and resources.
T1649	Steal or Forge Authentication Certificates	Adversaries may steal or forge certificates used for authentication to access remote systems or resources. Digital certificates are often used to sign and encrypt messages and/or files. Certificates are also used as authentication material. For example, Entra ID device certificates and Active Directory Certificate Services (AD CS) certificates bind to an identity and can be used as credentials for domain accounts.

T1558	Steal or Forge Kerberos Tickets	Adversaries may attempt to subvert Kerberos authentication by stealing or forging Kerberos tickets to enable Pass the Ticket. Kerberos is an authentication protocol widely used in modern Windows domain environments. In Kerberos environments, referred to as "realms", there are three basic participants: client, service, and Key Distribution Center (KDC). Clients request access to a service and through the exchange of Kerberos tickets, originating from KDC, they are granted access after having successfully authenticated. The KDC is responsible for both authentication and ticket granting. Adversaries may attempt to abuse Kerberos by stealing tickets or forging tickets to enable unauthorized access.
T1558.001	Golden Ticket	Adversaries who have the KRBTGT account password hash may forge Kerberos ticket-granting tickets (TGT), also known as a golden ticket. Golden tickets enable adversaries to generate authentication material for any account in Active Directory.
T1558.002	Silver Ticket	Adversaries who have the password hash of a target service account (e.g. SharePoint, MSSQL) may forge Kerberos ticket granting service (TGS) tickets, also known as silver tickets. Kerberos TGS tickets are also known as service tickets.
T1558.003	Kerberoasting	Adversaries may abuse a valid Kerberos ticket-granting ticket (TGT) or sniff network traffic to obtain a ticket-granting service (TGS) ticket that may be vulnerable to Brute Force.
T1558.004	AS-REP Roasting	Adversaries may reveal credentials of accounts that have disabled Kerberos preauthentication by Password Cracking Kerberos messages.
T1558.005	Ccache Files	Adversaries may attempt to steal Kerberos tickets stored in credential cache files (or ccache). These files are used for short term storage of a user's active session credentials. The ccache file is created upon user authentication and allows for access to multiple services without the user having to re-enter credentials.

T1539	Steal Web Session Cookie	An adversary may steal web application or service session cookies and use them to gain access to 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.
T1552	Unsecured Credentials	Adversaries may search compromised systems to find and obtain insecurely stored credentials. These credentials can be stored and/or misplaced in many locations on a system, including plaintext files (e.g. Bash History), operating system or application-specific repositories (e.g. Credentials in Registry), or other specialized files/artifacts (e.g. Private Keys).
T1552.001	Credentials In Files	Adversaries may search local file systems and remote file shares for files containing insecurely stored credentials. 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.
T1552.002	Credentials in Registry	Adversaries may search the Registry on compromised systems for insecurely stored credentials. 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.

T1552.003	Bash History	Adversaries may search the bash command history on compromised systems for insecurely stored credentials. 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 .bash_history file. For each user, this file resides at the same location: ~/.bash_history. 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. Adversaries can abuse this by looking through the file for potential credentials.
T1552.004	Private Keys	Adversaries may search for private key certificate files on compromised systems for insecurely stored credentials. Private cryptographic keys and certificates are used for authentication, encryption/decryption, and digital signatures. Common key and certificate file extensions include: .key, .pgp, .gpg, .ppk., .p12, .pem, .pfx, .cer, .p7b, .asc.
T1552.005	Cloud Instance Metadata API	Adversaries may attempt to access the Cloud Instance Metadata API to collect credentials and other sensitive data.
T1552.006	Group Policy Preferences	Adversaries may attempt to find unsecured credentials in Group Policy Preferences (GPP). GPP are tools that allow administrators to create domain policies with embedded credentials. These policies allow administrators to set local accounts.
T1552.007	Container API	Adversaries may gather credentials via APIs within a containers environment. APIs in these environments, such as the Docker API and Kubernetes APIs, allow a user to remotely manage their container resources and cluster components.

T1552.008	Chat Messages	Adversaries may directly collect unsecured credentials stored or passed through user communication services. Credentials may be sent and stored in user chat communication applications such as email, chat services like Slack or Teams, collaboration tools like Jira or Trello, and any other services that support user communication. Users may share various forms of credentials (such as usernames and passwords, API keys, or authentication tokens) on private or public corporate internal communications channels.

Tactic: Discovery
ID: TA0007
Description:
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-gatheringhe 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.

Techniques Under the Discovery Tactic

ID	Name	Description

T1087	Account Discovery	Adversaries may attempt to get a listing of valid accounts, usernames, or email addresses on a system or within a compromised environment. This information can help adversaries determine which accounts exist, which can aid in follow-on behavior such as brute-forcing, spear-phishing attacks, or account takeovers (e.g., Valid Accounts).
T1087.001	Local Account	Adversaries may attempt to get a listing of local system accounts. This information can help adversaries determine which local accounts exist on a system to aid in follow-on behavior.
T1087.002	Domain Account	Adversaries may attempt to get a listing of domain accounts. This information can help adversaries determine which domain accounts exist to aid in follow-on behavior such as targeting specific accounts which possess particular privileges.
T1087.003	Email Account	Adversaries may attempt to get a listing of email addresses and accounts. Adversaries may try to dump Exchange address lists such as global address lists (GALs).
T1087.004	Cloud Account	Adversaries may attempt to get a listing of cloud accounts. Cloud accounts are those created and configured by an organization for use by users, remote support, services, or for administration of resources within a cloud service provider or SaaS application.
T1010	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. For example, information about application windows could be used identify potential data to collect as well as identifying security tooling (Security Software Discovery) to evade.

T1217	Browser Information Discovery	Adversaries may enumerate information about browsers to learn more about compromised environments. Data saved by browsers (such as bookmarks, accounts, and browsing history) may reveal a variety of personal information about users (e.g., banking sites, relationships/interests, social media, etc.) as well as details about internal network resources such as servers, tools/dashboards, or other related infrastructure.
T1580	Cloud Infrastructure Discovery	An adversary may attempt to discover infrastructure and resources that are available within an infrastructure-as-a-service (IaaS) environment. This includes compute service resources such as instances, virtual machines, and snapshots as well as resources of other services including the storage and database services.
T1538	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.
T1526	Cloud Service Discovery	An adversary may attempt to enumerate the cloud services running on a system after gaining access. These methods can differ from platform-as-a-service (PaaS), to infrastructure-as-a-service (IaaS), or software-as-a-service (SaaS). Many services exist throughout the various cloud providers and can include Continuous Integration and Continuous Delivery (CI/CD), Lambda Functions, Entra ID, etc. They may also include security services, such as AWS GuardDuty and Microsoft Defender for Cloud, and logging services, such as AWS CloudTrail and Google Cloud Audit Logs.

T1619	Cloud Storage Object Discovery	Adversaries may enumerate objects in cloud storage infrastructure. Adversaries may use this information during automated discovery to shape follow-on behaviors, including requesting all or specific objects from cloud storage. Similar to File and Directory Discovery on a local host, after identifying available storage services (i.e. Cloud Infrastructure Discovery) adversaries may access the contents/objects stored in cloud infrastructure.
T1613	Container and Resource Discovery	Adversaries may attempt to discover containers and other resources that are available within a containers environment. Other resources may include images, deployments, pods, nodes, and other information such as the status of a cluster.
T1622	Debugger Evasion	Adversaries may employ various means to detect and avoid debuggers. Debuggers are typically used by defenders to trace and/or analyze the execution of potential malware payloads.
T1652	Device Driver Discovery	Adversaries may attempt to enumerate local device drivers on a victim host. Information about device drivers may highlight various insights that shape follow-on behaviors, such as the function/purpose of the host, present security tools (i.e. Security Software Discovery) or other defenses (e.g., Virtualization/Sandbox Evasion), as well as potential exploitable vulnerabilities (e.g., Exploitation for Privilege Escalation).

T1482	Domain Trust Discovery	Adversaries may attempt to gather information on domain trust relationships that may be used to identify lateral movement 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. 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, Pass the Ticket, and Kerberoasting. Domain trusts can be enumerated using the DSEnumerateDomainTrusts() Win32 API call, .NET methods, and LDAP. The Windows utility Nltest is known to be used by adversaries to enumerate domain trusts.
T1083	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 during automated discovery to shape follow-on behaviors, including whether or not the adversary fully infects the target and/or attempts specific actions.
T1615	Group Policy Discovery	Adversaries may gather information on Group Policy settings to identify paths for privilege escalation, security measures applied within a domain, and to discover patterns in domain objects that can be manipulated or used to blend in the environment. Group Policy allows for centralized management of user and computer settings in Active Directory (AD). Group policy objects (GPOs) are containers for group policy settings made up of files stored within a predictable network path \<DOMAIN>\SYSVOL\<DOMAIN>\Policies\.
T1654	Log Enumeration	Adversaries may enumerate system and service logs to find useful data. These logs may highlight various types of valuable insights for an adversary, such as user authentication records (Account Discovery), security or vulnerable software (Software Discovery), or hosts within a compromised network (Remote System Discovery).

T1046	Network Service Discovery	Adversaries may attempt to get a listing of services running on remote hosts and local network infrastructure devices, including those that may be vulnerable to remote software exploitation. Common methods to acquire this information include port and/or vulnerability scans using tools that are brought onto a system.
T1135	Network Share Discovery	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. Networks often contain shared network drives and folders that enable users to access file directories on various systems across a network.
T1040	Network Sniffing	Adversaries may passively sniff network traffic to capture information about an environment, including authentication material passed over the network. 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.
T1201	Password Policy Discovery	Adversaries may attempt to access detailed information about the password policy used within an enterprise network or cloud environment. Password policies are a way to enforce complex passwords that are difficult to guess or crack through Brute Force. This information may 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).

T1120	Peripheral Device Discovery	Adversaries may attempt to gather information about attached peripheral devices and components connected to a computer system. Peripheral devices could include auxiliary resources that support a variety of functionalities such as keyboards, printers, cameras, smart card readers, or removable storage. The information may be used to enhance their awareness of the system and network environment or may be used for further actions.
T1069	Permission Groups Discovery	Adversaries may attempt to discover group and permission settings. This information can help adversaries determine which user accounts and groups are available, the membership of users in particular groups, and which users and groups have elevated permissions.
T1069.001	Local Groups	Adversaries may attempt to find local system groups and permission settings. The knowledge of local system permission groups can help adversaries determine which groups exist and which users belong to a particular group. Adversaries may use this information to determine which users have elevated permissions, such as the users found within the local administrators group.
T1069.002	Domain Groups	Adversaries may attempt to find domain-level groups and permission settings. The knowledge of domain-level permission groups can help adversaries determine which groups exist and which users belong to a particular group. Adversaries may use this information to determine which users have elevated permissions, such as domain administrators.
T1069.003	Cloud Groups	Adversaries may attempt to find cloud groups and permission settings. The knowledge of cloud permission groups can help adversaries determine the particular roles of users and groups within an environment, as well as which users are associated with a particular group.

T1057	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/applications running on systems within the network. Administrator or otherwise elevated access may provide better process details. Adversaries may use the information from Process Discovery during automated discovery to shape follow-on behaviors, including whether or not the adversary fully infects the target and/or attempts specific actions.
T1012	Query Registry	Adversaries may interact with the Windows Registry to gather information about the system, configuration, and installed software.
T1018	Remote System Discovery	Adversaries may attempt to get a listing of other systems by IP address, hostname, or other logical identifier on a network that may be used for Lateral Movement from the current system. Functionality could exist within remote access tools to enable this, but utilities available on the operating system could also be used such as Ping or net view using Net.
T1518	Software Discovery	Adversaries may attempt to get a listing of software and software versions that are installed on a system or in a cloud environment. Adversaries may use the information from Software Discovery during automated discovery to shape follow-on behaviors, including whether or not the adversary fully infects the target and/or attempts specific actions.
T1518.001	Security Software Discovery	Adversaries may attempt to get a listing of security software, configurations, defensive tools, and sensors that are installed on a system or in a cloud environment. This may include things such as cloud monitoring agents and anti-virus. Adversaries may use the information from Security Software Discovery during automated discovery to shape follow-on behaviors, including whether or not the adversary fully infects the target and/or attempts specific actions.

T1082	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 during automated discovery to shape follow-on behaviors, including whether or not the adversary fully infects the target and/or attempts specific actions.
T1614	System Location Discovery	Adversaries may gather information in an attempt to calculate the geographical location of a victim host. Adversaries may use the information from System Location Discovery during automated discovery to shape follow-on behaviors, including whether or not the adversary fully infects the target and/or attempts specific actions.
T1614.001	System Language Discovery	Adversaries may attempt to gather information about the system language of a victim in order to infer the geographical location of that host. This information may be used to shape follow-on behaviors, including whether the adversary infects the target and/or attempts specific actions. This decision may be employed by malware developers and operators to reduce their risk of attracting the attention of specific law enforcement agencies or prosecution/scrutiny from other entities.
T1016	System Network Configuration Discovery	Adversaries may look for details about the network configuration and settings, such as IP and/or MAC addresses, of systems they access or through information discovery of remote systems. Several operating system administration utilities exist that can be used to gather this information. Examples include Arp, ipconfig/ifconfig, nbtstat, and route.
T1016.001	Internet Connection Discovery	Adversaries may check for Internet connectivity on compromised systems. This may be performed during automated discovery and can be accomplished in numerous ways such as using Ping, tracert, and GET requests to websites.

T1016.002	Wi-Fi Discovery	Adversaries may search for information about Wi-Fi networks, such as network names and passwords, on compromised systems. Adversaries may use Wi-Fi information as part of Account Discovery, Remote System Discovery, and other discovery or Credential Access activity to support both ongoing and future campaigns.
T1049	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.
T1033	System Owner/User Discovery	Adversaries may attempt to identify the primary user, currently logged in user, set of users that commonly uses a system, or whether a user is actively using the system. They may do this, for example, by retrieving account usernames or by using OS Credential Dumping. 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 during automated discovery to shape follow-on behaviors, including whether or not the adversary fully infects the target and/or attempts specific actions.
T1007	System Service Discovery	Adversaries may try to gather information about registered local system services. Adversaries may obtain information about services using tools as well as OS utility commands such as sc query, tasklist /svc, systemctl --type=service, and net start.
T1124	System Time Discovery	An adversary may gather the system time and/or time zone settings from a local or remote system. The system time is set and stored by services, such as the Windows Time Service on Windows or systemsetup on macOS. These time settings may also be synchronized between systems and services in an enterprise network, typically accomplished with a network time server within a domain.

T1497	Virtualization/Sandbox Evasion	Adversaries may employ various means to detect and avoid virtualization and analysis environments. This may include changing behaviors based on the results of checks for the presence of artifacts indicative of a virtual machine environment (VME) or sandbox. If the adversary detects a VME, they may alter their malware to disengage from the victim or conceal the core functions of the implant. They may also search for VME artifacts before dropping secondary or additional payloads. Adversaries may use the information learned from Virtualization/Sandbox Evasion during automated discovery to shape follow-on behaviors.
T1497.001	System Checks	Adversaries may employ various system checks to detect and avoid virtualization and analysis environments. This may include changing behaviors based on the results of checks for the presence of artifacts indicative of a virtual machine environment (VME) or sandbox. If the adversary detects a VME, they may alter their malware to disengage from the victim or conceal the core functions of the implant. They may also search for VME artifacts before dropping secondary or additional payloads. Adversaries may use the information learned from Virtualization/Sandbox Evasion during automated discovery to shape follow-on behaviors.
T1497.002	User Activity Based Checks	Adversaries may employ various user activity checks to detect and avoid virtualization and analysis environments. This may include changing behaviors based on the results of checks for the presence of artifacts indicative of a virtual machine environment (VME) or sandbox. If the adversary detects a VME, they may alter their malware to disengage from the victim or conceal the core functions of the implant. They may also search for VME artifacts before dropping secondary or additional payloads. Adversaries may use the information learned from Virtualization/Sandbox Evasion during automated discovery to shape follow-on behaviors.

T1497.003	Time Based Evasion	Adversaries may employ various time-based methods to detect and avoid virtualization and analysis environments. This may include enumerating time-based properties, such as uptime or the system clock, as well as the use of timers or other triggers to avoid a virtual machine environment (VME) or sandbox, specifically those that are automated or only operate for a limited amount of time.

Tactic: Lateral Movement

ID: TA0008

Description:

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.

Techniques Under the Lateral Movement Tactic

ID	Name	Description
T1210	Exploitation of Remote Services	Adversaries may exploit remote services to gain unauthorized access to internal systems once inside of a network. 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.

T1534	Internal Spearphishing	After they already have access to accounts or systems within the environment, adversaries may use internal spearphishing to gain access to additional information or compromise other users within the same organization. Internal spearphishing is multi-staged campaign where a legitimate account is initially compromised either by controlling the user's device or by compromising the account credentials of the user. Adversaries may then attempt to take advantage of the trusted internal account to increase the likelihood of tricking more victims into falling for phish attempts, often incorporating Impersonation.
T1570	Lateral Tool Transfer	Adversaries may transfer tools or other files between systems in a compromised environment. Once brought into the victim environment (i.e., Ingress Tool Transfer) files may then be copied from one system to another to stage adversary tools or other files over the course of an operation.
T1563	Remote Service Session Hijacking	Adversaries may take control of preexisting sessions with remote services to move laterally in an environment. Users may use valid credentials to log into a service specifically designed to accept remote connections, such as telnet, SSH, and RDP. When a user logs into a service, a session will be established that will allow them to maintain a continuous interaction with that service.
T1563.001	SSH Hijacking	Adversaries may hijack a legitimate user's SSH session to move laterally within an environment. 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.

T1563.002	RDP Hijacking	Adversaries may hijack a legitimate user’s remote desktop session to move laterally within an environment. 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).
T1021	Remote Services	Adversaries may use Valid Accounts to log into a service that accepts remote connections, such as telnet, SSH, and VNC. The adversary may then perform actions as the logged-on user.
T1021.001	Remote Desktop Protocol	Adversaries may use Valid Accounts to log into a computer using the Remote Desktop Protocol (RDP). The adversary may then perform actions as the logged-on user.
T1021.002	SMB/Windows Admin Shares	Adversaries may use Valid Accounts to interact with a remote network share using Server Message Block (SMB). The adversary may then perform actions as the logged-on user.
T1021.003	Distributed Component Object Model	Adversaries may use Valid Accounts to interact with remote machines by taking advantage of Distributed Component Object Model (DCOM). The adversary may then perform actions as the logged-on user.
T1021.004	SSH	Adversaries may use Valid Accounts to log into remote machines using Secure Shell (SSH). The adversary may then perform actions as the logged-on user.
T1021.005	VNC	Adversaries may use Valid Accounts to remotely control machines using Virtual Network Computing (VNC). VNC is a platform-independent desktop sharing system that uses the RFB ("remote framebuffer") protocol to enable users to remotely control another computer’s display by relaying the screen, mouse, and keyboard inputs over the network.
T1021.006	Windows Remote Management	Adversaries may use Valid Accounts to interact with remote systems using Windows Remote Management (WinRM). The adversary may then perform actions as the logged-on user.

T1021.007	Cloud Services	Adversaries may log into accessible cloud services within a compromised environment using Valid Accounts that are synchronized with or federated to on-premises user identities. The adversary may then perform management actions or access cloud-hosted resources as the logged-on user.
T1021.008	Direct Cloud VM Connections	Adversaries may leverage Valid Accounts to log directly into accessible cloud hosted compute infrastructure through cloud native methods. Many cloud providers offer interactive connections to virtual infrastructure that can be accessed through the Cloud API, such as Azure Serial Console, AWS EC2 Instance Connect, and AWS System Manager..
T1091	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.
T1072	Software Deployment Tools	Adversaries may gain access to and use centralized software suites installed within an enterprise to execute commands and move laterally through the network. Configuration management and software deployment applications may be used in an enterprise network or cloud environment for routine administration purposes. These systems may also be integrated into CI/CD pipelines. Examples of such solutions include: SCCM, HBSS, Altiris, AWS Systems Manager, Microsoft Intune, Azure Arc, and GCP Deployment Manager.

T1080	Taint Shared Content	Adversaries may deliver payloads to remote systems by adding content to shared storage locations, such as network drives or internal code repositories. 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.
T1550	Use Alternate Authentication Material	Adversaries may use alternate authentication material, such as password hashes, Kerberos tickets, and application access tokens, in order to move laterally within an environment and bypass normal system access controls.
T1550.001	Application Access Token	Adversaries may use stolen 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 or services and used in lieu of login credentials.
T1550.002	Pass the Hash	Adversaries may "pass the hash" using stolen password hashes to move laterally within an environment, bypassing normal system access controls. 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.
T1550.003	Pass the Ticket	Adversaries may "pass the ticket" using stolen Kerberos tickets to move laterally within an environment, bypassing normal system access controls. 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.

T1550.004	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.

Tactic: Collection
ID: TA0009
Description:
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 either steal (exfiltrate) the data or to use the data to gain more information about the target environment. Common target sources include various drive types, browsers, audio, video, and email. Common collection methods include capturing screenshots and keyboard input.

Techniques Under the Collection Tactic

ID	Name	Description
T1557	Adversary-in-the-Middle	Adversaries may attempt to position themselves between two or more networked devices using an adversary-in-the-middle (AiTM) technique to support follow-on behaviors such as Network Sniffing, Transmitted Data Manipulation, or replay attacks (Exploitation for Credential Access). By abusing features of common networking protocols that can determine the flow of network traffic (e.g. ARP, DNS, LLMNR, etc.), adversaries may force a device to communicate through an adversary controlled system so they can collect information or perform additional actions.
T1557.001	LLMNR/NBT-NS Poisoning and SMB Relay	By responding to LLMNR/NBT-NS network traffic, adversaries may spoof an authoritative source for name resolution to force communication with an adversary controlled system. This activity may be used to collect or relay authentication materials.

T1557.002	ARP Cache Poisoning	Adversaries may poison Address Resolution Protocol (ARP) caches to position themselves between the communication of two or more networked devices. This activity may be used to enable follow-on behaviors such as Network Sniffing or Transmitted Data Manipulation.
T1557.003	DHCP Spoofing	Adversaries may redirect network traffic to adversary-owned systems by spoofing Dynamic Host Configuration Protocol (DHCP) traffic and acting as a malicious DHCP server on the victim network. By achieving the adversary-in-the-middle (AiTM) position, adversaries may collect network communications, including passed credentials, especially those sent over insecure, unencrypted protocols. This may also enable follow-on behaviors such as Network Sniffing or Transmitted Data Manipulation.
T1557.004	Evil Twin	Adversaries may host seemingly genuine Wi-Fi access points to deceive users into connecting to malicious networks as a way of supporting follow-on behaviors such as Network Sniffing, Transmitted Data Manipulation, or Input Capture.
T1560	Archive Collected Data	An adversary may compress and/or encrypt data that is collected prior to exfiltration. Compressing the data can help to obfuscate the collected data and minimize the amount of data sent over the network. Encryption can be used to hide information that is being exfiltrated from detection or make exfiltration less conspicuous upon inspection by a defender.
T1560.001	Archive via Utility	Adversaries may use utilities to compress and/or encrypt collected data prior to exfiltration. Many utilities include functionalities to compress, encrypt, or otherwise package data into a format that is easier/more secure to transport.
T1560.002	Archive via Library	An adversary may compress or encrypt data that is collected prior to exfiltration using 3rd party libraries. Many libraries exist that can archive data, including Python rarfile , libzip , and zlib . Most libraries include functionality to encrypt and/or compress data.

T1560.003	Archive via Custom Method	An adversary may compress or encrypt data that is collected prior to exfiltration using a custom method. Adversaries may choose to use custom archival methods, such as encryption with XOR or stream ciphers implemented with no external library or utility references. Custom implementations of well-known compression algorithms have also been used.
T1123	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.
T1119	Automated Collection	Once established within a system or network, an adversary may use automated techniques for collecting internal data. Methods for performing this technique could include use of a Command and Scripting Interpreter to search for and copy information fitting set criteria such as file type, location, or name at specific time intervals.
T1185	Browser Session Hijacking	Adversaries may take advantage of security vulnerabilities and inherent functionality in browser software to change content, modify user-behaviors, and intercept information as part of various browser session hijacking techniques.
T1115	Clipboard Data	Adversaries may collect data stored in the clipboard from users copying information within or between applications.
T1530	Data from Cloud Storage	Adversaries may access data from cloud storage.
T1602	Data from Configuration Repository	Adversaries may collect data related to managed devices from configuration repositories. Configuration repositories are used by management systems in order to configure, manage, and control data on remote systems. Configuration repositories may also facilitate remote access and administration of devices.
T1602.001	SNMP (MIB Dump)	Adversaries may target the Management Information Base (MIB) to collect and/or mine valuable information in a network managed using Simple Network Management Protocol (SNMP).

T1602.002	Network Device Configuration Dump	Adversaries may access network configuration files to collect sensitive data about the device and the network. The network configuration is a file containing parameters that determine the operation of the device. The device typically stores an in-memory copy of the configuration while operating, and a separate configuration on non-volatile storage to load after device reset. Adversaries can inspect the configuration files to reveal information about the target network and its layout, the network device and its software, or identifying legitimate accounts and credentials for later use.
T1213	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, such as Credential Access, Lateral Movement, or Defense Evasion, or direct access to the target information. Adversaries may also abuse external sharing features to share sensitive documents with recipients outside of the organization (i.e., Transfer Data to Cloud Account).
T1213.001	Confluence	Adversaries may leverage Confluence repositories to mine valuable information. Often found in development environments alongside Atlassian JIRA, Confluence is generally used to store development-related documentation, however, in general may contain more diverse categories of useful information, such as:
T1213.002	Sharepoint	Adversaries may leverage the SharePoint repository as a source to mine valuable information. SharePoint will often contain useful information for an adversary to learn about the structure and functionality of the internal network and systems. For example, the following is a list of example information that may hold potential value to an adversary and may also be found on SharePoint:

T1213.003	Code Repositories	Adversaries may leverage code repositories to collect valuable information. Code repositories are tools/services that store source code and automate software builds. They may be hosted internally or privately on third party sites such as Github, GitLab, SourceForge, and BitBucket. Users typically interact with code repositories through a web application or command-line utilities such as git.
T1213.004	Customer Relationship Management Software	Adversaries may leverage Customer Relationship Management (CRM) software to mine valuable information. CRM software is used to assist organizations in tracking and managing customer interactions, as well as storing customer data.
T1213.005	Messaging Applications	Adversaries may leverage chat and messaging applications, such as Microsoft Teams, Google Chat, and Slack, to mine valuable information.
T1005	Data from Local System	Adversaries may search local system sources, such as file systems and configuration files or local databases, to find files of interest and sensitive data prior to Exfiltration.
T1039	Data from Network Shared Drive	Adversaries may search network shares on computers they have compromised to find files of interest. 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. Interactive command shells may be in use, and common functionality within cmd may be used to gather information.
T1025	Data from Removable Media	Adversaries may search connected removable media on computers they have compromised to find files of interest. Sensitive data can be collected from any removable media (optical disk drive, USB memory, etc.) connected to the compromised system prior to Exfiltration. Interactive command shells may be in use, and common functionality within cmd may be used to gather information.

T1074	Data Staged	Adversaries may stage collected data 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 Archive Collected Data. Interactive command shells may be used, and common functionality within cmd and bash may be used to copy data into a staging location.
T1074.001	Local Data Staging	Adversaries may stage collected data in a central location or directory on the local system prior to Exfiltration. Data may be kept in separate files or combined into one file through techniques such as Archive Collected Data. Interactive command shells may be used, and common functionality within cmd and bash may be used to copy data into a staging location.
T1074.002	Remote Data Staging	Adversaries may stage data collected from multiple systems in a central location or directory on one system prior to Exfiltration. Data may be kept in separate files or combined into one file through techniques such as Archive Collected Data. Interactive command shells may be used, and common functionality within cmd and bash may be used to copy data into a staging location.
T1114	Email Collection	Adversaries may target user email to collect sensitive information. Emails may contain sensitive data, including trade secrets or personal information, that can prove valuable to adversaries. Emails may also contain details of ongoing incident response operations, which may allow adversaries to adjust their techniques in order to maintain persistence or evade defenses. Adversaries can collect or forward email from mail servers or clients.
T1114.001	Local Email Collection	Adversaries may target user email on local systems to collect sensitive information. Files containing email data can be acquired from a user's local system, such as Outlook storage or cache files.

T1114.002	Remote Email Collection	Adversaries may target an Exchange server, Office 365, or Google Workspace to collect sensitive information. 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, Office 365, or Google Workspace to access email using credentials or access tokens. Tools such as MailSniper can be used to automate searches for specific keywords.
T1114.003	Email Forwarding Rule	Adversaries may setup email forwarding rules to collect sensitive information. Adversaries may 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. Furthermore, email forwarding rules can allow adversaries to maintain persistent access to victim's emails even after compromised credentials are reset by administrators. Most email clients allow users to create inbox rules for various email functions, including forwarding to a different recipient. These rules may be created through a local email application, a web interface, or by command-line interface. 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.
T1056	Input Capture	Adversaries may use methods of capturing user input to obtain credentials or collect information. During normal system usage, users often provide credentials to various different locations, such as login pages/portals or system dialog boxes. Input capture mechanisms may be transparent to the user (e.g. Credential API Hooking) or rely on deceiving the user into providing input into what they believe to be a genuine service (e.g. Web Portal Capture).

T1056.001	Keylogging	Adversaries may log user keystrokes to intercept credentials as the user types them. Keylogging is likely to be used to acquire credentials for new access opportunities when OS Credential Dumping efforts are not effective, and may require an adversary to intercept keystrokes on a system for a substantial period of time before credentials can be successfully captured. In order to increase the likelihood of capturing credentials quickly, an adversary may also perform actions such as clearing browser cookies to force users to reauthenticate to systems.
T1056.002	GUI Input Capture	Adversaries may mimic common operating system GUI components to prompt users for credentials with a seemingly legitimate 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).
T1056.003	Web Portal Capture	Adversaries may 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. For example, a compromised login page may log provided user credentials before logging the user in to the service.
T1056.004	Credential API Hooking	Adversaries may hook into Windows application programming interface (API) functions to collect user credentials. Malicious hooking mechanisms may capture API calls that include parameters that reveal user authentication credentials. Unlike Keylogging, this technique focuses specifically on API functions that include parameters that reveal user credentials. Hooking involves redirecting calls to these functions and can be implemented via:

T1113	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. Taking a screenshot is also typically possible through native utilities or API calls, such as CopyFromScreen, xwd, or screencapture.
T1125	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.

Tactic: Command and Control
ID: TA0011
Description:
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.

Techniques Under the Command and Control Tactic

ID	Name	Description
T1071	Application Layer Protocol	Adversaries may communicate using OSI application layer protocols to avoid detection/network filtering by blending in with existing traffic. Commands to the remote system, and often the results of those commands, will be embedded within the protocol traffic between the client and server.

T1071.001	Web Protocols	Adversaries may communicate using application layer protocols associated with web traffic to avoid detection/network filtering by blending in with existing traffic. Commands to the remote system, and often the results of those commands, will be embedded within the protocol traffic between the client and server.
T1071.002	File Transfer Protocols	Adversaries may communicate using application layer protocols associated with transferring files to avoid detection/network filtering by blending in with existing traffic. Commands to the remote system, and often the results of those commands, will be embedded within the protocol traffic between the client and server.
T1071.003	Mail Protocols	Adversaries may communicate using application layer protocols associated with electronic mail delivery to avoid detection/network filtering by blending in with existing traffic. Commands to the remote system, and often the results of those commands, will be embedded within the protocol traffic between the client and server.
T1071.004	DNS	Adversaries may communicate using the Domain Name System (DNS) application layer protocol to avoid detection/network filtering by blending in with existing traffic. Commands to the remote system, and often the results of those commands, will be embedded within the protocol traffic between the client and server.
T1071.005	Publish/Subscribe Protocols	Adversaries may communicate using publish/subscribe (pub/sub) application layer protocols to avoid detection/network filtering by blending in with existing traffic. Commands to the remote system, and often the results of those commands, will be embedded within the protocol traffic between the client and server.

T1092	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. Commands and files would be relayed from the disconnected system to the Internet-connected system to which the adversary has direct access.
T1659	Content Injection	Adversaries may gain access and continuously communicate with victims by injecting malicious content into systems through online network traffic. Rather than luring victims to malicious payloads hosted on a compromised website (i.e., Drive-by Target followed by Drive-by Compromise), adversaries may initially access victims through compromised data-transfer channels where they can manipulate traffic and/or inject their own content. These compromised online network channels may also be used to deliver additional payloads (i.e., Ingress Tool Transfer) and other data to already compromised systems.
T1132	Data Encoding	Adversaries may encode data to make the content of command and control traffic more difficult to detect. Command and control (C2) information can be encoded using a standard data encoding system. Use of data encoding may adhere to existing protocol specifications and includes use of ASCII, Unicode, Base64, MIME, or other binary-to-text and character encoding systems. Some data encoding systems may also result in data compression, such as gzip.

T1132.001	Standard Encoding	Adversaries may encode data with a standard data encoding system to make the content of command and control traffic more difficult to detect. Command and control (C2) information can be encoded using a standard data encoding system that adheres to existing protocol specifications. Common data encoding schemes include ASCII, Unicode, hexadecimal, Base64, and MIME. Some data encoding systems may also result in data compression, such as gzip.
T1132.002	Non-Standard Encoding	Adversaries may encode data with a non-standard data encoding system to make the content of command and control traffic more difficult to detect. Command and control (C2) information can be encoded using a non-standard data encoding system that diverges from existing protocol specifications. Non-standard data encoding schemes may be based on or related to standard data encoding schemes, such as a modified Base64 encoding for the message body of an HTTP request.
T1001	Data Obfuscation	Adversaries may obfuscate command and control traffic to make it more difficult to detect. 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, or impersonating legitimate protocols.
T1001.001	Junk Data	Adversaries may add junk data to protocols used for command and control to make detection more difficult. By adding random or meaningless data to the protocols used for command and control, adversaries can prevent trivial methods for decoding, deciphering, or otherwise analyzing the traffic. Examples may include appending/prepending data with junk characters or writing junk characters between significant characters.

T1001.002	Steganography	Adversaries may use steganographic techniques to hide command and control traffic to make detection efforts more difficult. Steganographic techniques can be used to hide data in digital messages that are transferred between systems. This hidden information can be used for command and control of compromised systems. In some cases, the passing of files embedded using steganography, such as image or document files, can be used for command and control.
T1001.003	Protocol or Service Impersonation	Adversaries may impersonate legitimate protocols or web service traffic to disguise command and control activity and thwart analysis efforts. By impersonating legitimate protocols or web services, adversaries can make their command and control traffic blend in with legitimate network traffic.
T1568	Dynamic Resolution	Adversaries may dynamically establish connections to command and control infrastructure to evade common detections and remediations. This may be achieved by using malware that shares a common algorithm with the infrastructure the adversary uses to receive the malware's communications. These calculations can be used to dynamically adjust parameters such as the domain name, IP address, or port number the malware uses for command and control.
T1568.001	Fast Flux DNS	Adversaries may use Fast Flux DNS to hide a command and control channel behind an array of rapidly changing IP addresses linked to a single domain resolution. This technique uses a fully qualified domain name, with multiple IP addresses assigned to it which are swapped with high frequency, using a combination of round robin IP addressing and short Time-To-Live (TTL) for a DNS resource record.

T1568.002	Domain Generation Algorithms	Adversaries may make use of Domain Generation Algorithms (DGAs) to dynamically identify a destination domain 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 to block, track, or take over the command and control channel, as there potentially could be thousands of domains that malware can check for instructions.
T1568.003	DNS Calculation	Adversaries may perform calculations on addresses returned in DNS results to determine which port and IP address to use for command and control, rather than relying on a predetermined port number or the actual returned IP address. A IP and/or port number calculation can be used to bypass egress filtering on a C2 channel.
T1573	Encrypted Channel	Adversaries may employ an 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 secret keys are encoded and/or generated within malware samples/configuration files.
T1573.001	Symmetric Cryptography	Adversaries may employ a known symmetric encryption algorithm to conceal command and control traffic rather than relying on any inherent protections provided by a communication protocol. Symmetric encryption algorithms use the same key for plaintext encryption and ciphertext decryption. Common symmetric encryption algorithms include AES, DES, 3DES, Blowfish, and RC4.

T1573.002	Asymmetric Cryptography	Adversaries may employ a known asymmetric encryption algorithm to conceal command and control traffic rather than relying on any inherent protections provided by a communication protocol. Asymmetric cryptography, also known as public key cryptography, uses a keypair per party: one public that can be freely distributed, and one private. Due to how the keys are generated, the sender encrypts data with the receiver’s public key and the receiver decrypts the data with their private key. This ensures that only the intended recipient can read the encrypted data. Common public key encryption algorithms include RSA and ElGamal.
T1008	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.
T1665	Hide Infrastructure	Adversaries may manipulate network traffic in order to hide and evade detection of their C2 infrastructure. This can be accomplished in various ways including by identifying and filtering traffic from defensive tools, masking malicious domains to obfuscate the true destination from both automated scanning tools and security researchers, and otherwise hiding malicious artifacts to delay discovery and prolong the effectiveness of adversary infrastructure that could otherwise be identified, blocked, or taken down entirely.
T1105	Ingress Tool Transfer	Adversaries may transfer tools or other files from an external system into a compromised environment. Tools or files may be copied from an external adversary-controlled system to the victim network through the command and control channel or through alternate protocols such as ftp. Once present, adversaries may also transfer/spread tools between victim devices within a compromised environment (i.e. Lateral Tool Transfer).

T1104	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.
T1095	Non-Application Layer Protocol	Adversaries may use an OSI 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. 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).
T1571	Non-Standard Port	Adversaries may communicate using a protocol and port pairing that are typically not associated. For example, HTTPS over port 8088 or port 587 as opposed to the traditional port 443. Adversaries may make changes to the standard port used by a protocol to bypass filtering or muddle analysis/parsing of network data.
T1572	Protocol Tunneling	Adversaries may tunnel network communications to and from a victim system within a separate protocol to avoid detection/network filtering and/or enable access to otherwise unreachable systems. Tunneling involves explicitly encapsulating a protocol within another. This behavior may conceal malicious traffic by blending in with existing traffic and/or provide an outer layer of encryption (similar to a VPN). Tunneling could also enable routing of network packets that would otherwise not reach their intended destination, such as SMB, RDP, or other traffic that would be filtered by network appliances or not routed over the Internet.

T1090	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, ZXProxy, and ZXPortMap. Adversaries use these types of proxies to manage command and control communications, reduce the number of simultaneous outbound network connections, provide resiliency in the face of connection loss, or to ride over existing trusted communications paths between victims to avoid suspicion. Adversaries may chain together multiple proxies to further disguise the source of malicious traffic.
T1090.001	Internal Proxy	Adversaries may use an internal proxy to direct command and control traffic between two or more systems in a compromised environment. Many tools exist that enable traffic redirection through proxies or port redirection, including HTRAN, ZXProxy, and ZXPortMap. Adversaries use internal proxies to manage command and control communications inside a compromised environment, 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 infected systems to avoid suspicion. Internal proxy connections may use common peer-to-peer (p2p) networking protocols, such as SMB, to better blend in with the environment.
T1090.002	External Proxy	Adversaries may use an external proxy to 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, ZXProxy, and ZXPortMap. Adversaries use these types of proxies to manage command and control communications, to provide resiliency in the face of connection loss, or to ride over existing trusted communications paths to avoid suspicion.

T1090.003	Multi-hop Proxy	Adversaries may chain together multiple proxies to disguise the source of malicious traffic. 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.
T1090.004	Domain Fronting	Adversaries may take 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. Domain fronting 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).
T1219	Remote Access Software	An adversary may use legitimate desktop support and remote access software to establish an interactive command and control channel to target systems within networks. These services, such as VNC, Team Viewer, AnyDesk, ScreenConnect, LogMein, AmmyyAdmin, and other remote monitoring and management (RMM) tools, are commonly used as legitimate technical support software and may be allowed by application control within a target environment.

T1205	Traffic Signaling	Adversaries may use traffic signaling to hide open ports or other malicious functionality used for persistence or command and control. Traffic signaling involves the use of a magic value or sequence that must be sent to a system to trigger a special response, such as opening a closed port or executing a malicious task. This may take the form of sending a series of packets with certain characteristics before a port will be opened that the adversary can use for command and control. Usually this series of packets consists of attempted connections to a predefined sequence of closed ports (i.e. Port Knocking), but can involve unusual flags, specific strings, or other unique characteristics. After the sequence is completed, opening a port may be accomplished by the host-based firewall, but could also be implemented by custom software.
T1205.001	Port Knocking	Adversaries may use port knocking to hide open ports used for persistence or command and control. To enable a port, an adversary sends a series of attempted connections to a predefined sequence of closed ports. After the sequence is completed, opening a port is often accomplished by the host based firewall, but could also be implemented by custom software.
T1205.002	Socket Filters	Adversaries may attach filters to a network socket to monitor then activate backdoors used for persistence or command and control. With elevated permissions, adversaries can use features such as the libpcap library to open sockets and install filters to allow or disallow certain types of data to come through the socket. The filter may apply to all traffic passing through the specified network interface (or every interface if not specified). When the network interface receives a packet matching the filter criteria, additional actions can be triggered on the host, such as activation of a reverse shell.

T1102	Web Service	Adversaries may use an existing, legitimate external Web service as a means for relaying data to/from a compromised system. Popular websites, cloud services, 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, Microsoft, 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.
T1102.001	Dead Drop Resolver	Adversaries may use an existing, legitimate external Web service to host information that points to additional 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.
T1102.002	Bidirectional Communication	Adversaries may use an existing, legitimate external Web service as a means for sending commands to and receiving output from a compromised system over the Web service channel. Compromised systems may leverage popular websites and social media to host command and control (C2) instructions. Those infected systems can then send the output from those commands back over that Web service channel. The return traffic may occur in a variety of ways, depending on the Web service being utilized. For example, the return traffic may take the form of the compromised system posting a comment on a forum, issuing a pull request to development project, updating a document hosted on a Web service, or by sending a Tweet.
T1102.003	One-Way Communication	Adversaries may use an existing, legitimate external Web service as a means for sending commands to a compromised system without receiving return output over the Web service channel. Compromised systems may leverage popular websites and social media to host command and control (C2) instructions. Those infected systems may opt to send the output from those commands back over a different C2 channel, including to another distinct Web service. Alternatively, compromised systems may return no output at all in cases where adversaries want to send instructions to systems and do not want a response.

Tactic: Exfiltration
ID: TA0010
Description:
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.

Techniques Under the Exfiltration Tactic

ID	Name	Description
T1020	Automated Exfiltration	Adversaries may exfiltrate data, such as sensitive documents, through the use of automated processing after being gathered during Collection.
T1020.001	Traffic Duplication	Adversaries may leverage traffic mirroring in order to automate data exfiltration over compromised infrastructure. Traffic mirroring is a native feature for some devices, often used for network analysis. For example, devices may be configured to forward network traffic to one or more destinations for analysis by a network analyzer or other monitoring device.
T1030	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.
T1048	Exfiltration Over Alternative Protocol	Adversaries may steal data by exfiltrating it over a different protocol than that of the existing command and control channel. The data may also be sent to an alternate network location from the main command and control server.

T1048.001	Exfiltration Over Symmetric Encrypted Non-C2 Protocol	Adversaries may steal data by exfiltrating it over a symmetrically encrypted network protocol other than that of the existing command and control channel. The data may also be sent to an alternate network location from the main command and control server.
T1048.002	Exfiltration Over Asymmetric Encrypted Non-C2 Protocol	Adversaries may steal data by exfiltrating it over an asymmetrically encrypted network protocol other than that of the existing command and control channel. The data may also be sent to an alternate network location from the main command and control server.
T1048.003	Exfiltration Over Unencrypted Non-C2 Protocol	Adversaries may steal data by exfiltrating it over an un-encrypted network protocol other than that of the existing command and control channel. The data may also be sent to an alternate network location from the main command and control server.
T1041	Exfiltration Over C2 Channel	Adversaries may steal data by exfiltrating it over an existing command and control channel. Stolen data is encoded into the normal communications channel using the same protocol as command and control communications.
T1011	Exfiltration Over Other Network Medium	Adversaries may attempt to exfiltrate data 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.
T1011.001	Exfiltration Over Bluetooth	Adversaries may attempt to exfiltrate data over Bluetooth rather than the command and control channel. If the command and control network is a wired Internet connection, an adversary may opt to exfiltrate data using a Bluetooth communication channel.

T1052	Exfiltration Over Physical Medium	Adversaries may attempt to exfiltrate data via a physical medium, such as a removable drive. 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.
T1052.001	Exfiltration over USB	Adversaries may attempt to exfiltrate data over a USB connected physical device. In certain circumstances, such as an air-gapped network compromise, exfiltration could occur via a USB device introduced by a user. The USB device could be used as the final exfiltration point or to hop between otherwise disconnected systems.
T1567	Exfiltration Over Web Service	Adversaries may use an existing, legitimate external Web service to exfiltrate data rather than their primary command and control channel. Popular Web services acting as an exfiltration mechanism may give a significant amount of cover due to the likelihood that hosts within a network are already communicating with them prior to compromise. Firewall rules may also already exist to permit traffic to these services.
T1567.001	Exfiltration to Code Repository	Adversaries may exfiltrate data to a code repository rather than over their primary command and control channel. Code repositories are often accessible via an API (ex: https://api.github.com). Access to these APIs are often over HTTPS, which gives the adversary an additional level of protection.
T1567.002	Exfiltration to Cloud Storage	Adversaries may exfiltrate data to a cloud storage service rather than over their primary command and control channel. Cloud storage services allow for the storage, edit, and retrieval of data from a remote cloud storage server over the Internet.

T1567.003	Exfiltration to Text Storage Sites	Adversaries may exfiltrate data to text storage sites instead of their primary command and control channel. Text storage sites, such as pastebin[.]com, are commonly used by developers to share code and other information.
T1567.004	Exfiltration Over Webhook	Adversaries may exfiltrate data to a webhook endpoint rather than over their primary command and control channel. Webhooks are simple mechanisms for allowing a server to push data over HTTP/S to a client without the need for the client to continuously poll the server. Many public and commercial services, such as Discord, Slack, and webhook.site, support the creation of webhook endpoints that can be used by other services, such as Github, Jira, or Trello. When changes happen in the linked services (such as pushing a repository update or modifying a ticket), these services will automatically post the data to the webhook endpoint for use by the consuming application.
T1029	Scheduled Transfer	Adversaries may schedule data exfiltration to 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.
T1537	Transfer Data to Cloud Account	Adversaries may exfiltrate data by transferring the data, including through sharing/syncing and creating backups of cloud environments, to another cloud account they control on the same service.

Tactic: Impact
ID: TA0040
Description:
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.

Techniques Under the Impact Tactic

ID	Name	Description
T1531	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 perform a System Shutdown/Reboot to set malicious changes into place.
T1485	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. Common operating system file deletion commands such as del and rm 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 and Disk Structure Wipe because individual files are destroyed rather than sections of a storage disk or the disk's logical structure.
T1485.001	Lifecycle-Triggered Deletion	Adversaries may modify the lifecycle policies of a cloud storage bucket to destroy all objects stored within.
T1486	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.

T1565	Data Manipulation	Adversaries may insert, delete, or manipulate data in order to influence external outcomes or hide activity, thus threatening the integrity of the data. By manipulating data, adversaries may attempt to affect a business process, organizational understanding, or decision making.
T1565.001	Stored Data Manipulation	Adversaries may insert, delete, or manipulate data at rest in order to influence external outcomes or hide activity, thus threatening the integrity of the data. By manipulating stored data, adversaries may attempt to affect a business process, organizational understanding, and decision making.
T1565.002	Transmitted Data Manipulation	Adversaries may alter data en route to storage or other systems in order to manipulate external outcomes or hide activity, thus threatening the integrity of the data. By manipulating transmitted data, adversaries may attempt to affect a business process, organizational understanding, and decision making.
T1565.003	Runtime Data Manipulation	Adversaries may modify systems in order to manipulate the data as it is accessed and displayed to an end user, thus threatening the integrity of the data. By manipulating runtime data, adversaries may attempt to affect a business process, organizational understanding, and decision making.
T1491	Defacement	Adversaries may modify visual content available internally or externally to an enterprise network, thus affecting the integrity of the original content. Reasons for Defacement include delivering messaging, intimidation, or claiming (possibly false) credit for an intrusion. 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.

T1491.001	Internal Defacement	An adversary may deface systems internal to an organization in an attempt to intimidate or mislead users, thus discrediting the integrity of the systems. This may take the form of modifications to internal websites, or directly to user systems with the replacement of the desktop wallpaper. Disturbing or offensive images may be used as a part of Internal Defacement in order to cause user discomfort, or to pressure compliance with accompanying messages. Since internally defacing systems exposes an adversary's presence, it often takes place after other intrusion goals have been accomplished.
T1491.002	External Defacement	An adversary may deface systems external to an organization in an attempt to deliver messaging, intimidate, or otherwise mislead an organization or users. External Defacement may ultimately cause users to distrust the systems and to question/discredit the system's integrity. Externally-facing websites are a common victim of defacement; often targeted by adversary and hacktivist groups in order to push a political message or spread propaganda. External 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.
T1561	Disk Wipe	Adversaries may wipe or corrupt raw disk data on specific systems or in large numbers in a network to interrupt availability to system and network resources. With direct write access to a disk, adversaries may attempt to overwrite portions of disk data. Adversaries may opt to wipe arbitrary portions of disk data and/or wipe disk structures like the master boot record (MBR). A complete wipe of all disk sectors may be attempted.
T1561.001	Disk Content Wipe	Adversaries may erase the contents of storage devices on specific systems or in large numbers in a network to interrupt availability to system and network resources.

T1561.002	Disk Structure Wipe	Adversaries may corrupt or wipe the disk data structures on a hard drive necessary to boot a system; targeting specific critical systems or in large numbers in a network to interrupt availability to system and network resources.
T1499	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 and to support other malicious activities, including distraction, hacktivism, and extortion.
T1499.001	OS Exhaustion Flood	Adversaries may launch a denial of service (DoS) attack targeting an endpoint's operating system (OS). A system's OS is responsible for managing the finite resources as well as preventing the entire system from being overwhelmed by excessive demands on its capacity. These attacks do not need to exhaust the actual resources on a system; the attacks may simply exhaust the limits and available resources that an OS self-imposes.
T1499.002	Service Exhaustion Flood	Adversaries may target the different network services provided by systems to conduct a denial of service (DoS). Adversaries often target the availability of DNS and web services, however others have been targeted as well. 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.
T1499.003	Application Exhaustion Flood	Adversaries may target resource intensive features of applications to cause a denial of service (DoS), denying availability to those applications. For example, specific features in web applications may be highly resource intensive. Repeated requests to those features may be able to exhaust system resources and deny access to the application or the server itself.

T1499.004	Application or System Exploitation	Adversaries may exploit software vulnerabilities that can cause an application or system to crash and deny availability to users. Some systems may automatically restart critical applications and services when crashes occur, but they can likely be re-exploited to cause a persistent denial of service (DoS) condition.
T1657	Financial Theft	Adversaries may steal monetary resources from targets through extortion, social engineering, technical theft, or other methods aimed at their own financial gain at the expense of the availability of these resources for victims. Financial theft is the ultimate objective of several popular campaign types including extortion by ransomware, business email compromise (BEC) and fraud, "pig butchering," bank hacking, and exploiting cryptocurrency networks.
T1495	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, thus denying the availability to use the devices and/or the system. 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 may include the motherboard, hard drive, or video cards.
T1490	Inhibit System Recovery	Adversaries may delete or remove built-in data and turn off services designed to aid in the recovery of a corrupted system to prevent recovery. This may deny access to available backups and recovery options.
T1498	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 and to support other malicious activities, including distraction, hacktivism, and extortion.

T1498.001	Direct Network Flood	Adversaries may attempt to cause a denial of service (DoS) by directly sending a high-volume of network traffic to a target. This DoS attack may also reduce the availability and functionality of the targeted system(s) and network. 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 flooding. Stateless protocols such as UDP or ICMP are commonly used but stateful protocols such as TCP can be used as well.
T1498.002	Reflection Amplification	Adversaries may attempt to cause a denial of service (DoS) by reflecting a high-volume of network traffic to a target. 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 reflectors may be used to focus traffic on the target. This Network DoS attack may also reduce the availability and functionality of the targeted system(s) and network.
T1496	Resource Hijacking	Adversaries may leverage the resources of co-opted systems to complete resource-intensive tasks, which may impact system and/or hosted service availability.
T1496.001	Compute Hijacking	Adversaries may leverage the compute resources of co-opted systems to complete resource-intensive tasks, which may impact system and/or hosted service availability.
T1496.002	Bandwidth Hijacking	Adversaries may leverage the network bandwidth resources of co-opted systems to complete resource-intensive tasks, which may impact system and/or hosted service availability.

T1496.003	SMS Pumping	Adversaries may leverage messaging services for SMS pumping, which may impact system and/or hosted service availability. SMS pumping is a type of telecommunications fraud whereby a threat actor first obtains a set of phone numbers from a telecommunications provider, then leverages a victim’s messaging infrastructure to send large amounts of SMS messages to numbers in that set. By generating SMS traffic to their phone number set, a threat actor may earn payments from the telecommunications provider.
T1496.004	Cloud Service Hijacking	Adversaries may leverage compromised software-as-a-service (SaaS) applications to complete resource-intensive tasks, which may impact hosted service availability.
T1489	Service Stop	Adversaries may stop or disable services on a system to render those services unavailable to legitimate users. Stopping critical services or processes can inhibit or stop response to an incident or aid in the adversary's overall objectives to cause damage to the environment.
T1529	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 or network device. In some cases, these commands may also be used to initiate a shutdown/reboot of a remote computer or network device via Network Device CLI (e.g. reload).