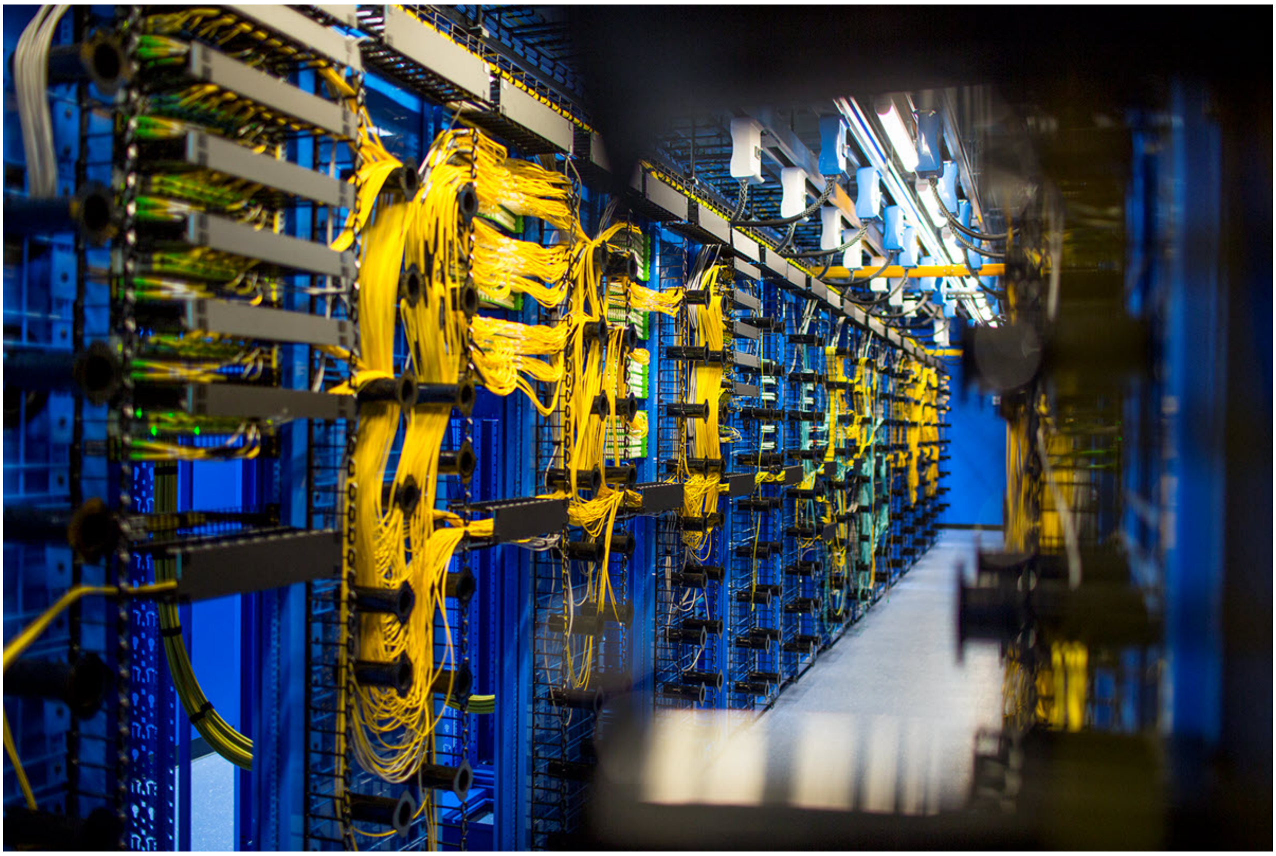
New “Prestige” ransomware impacts organizations in Ukraine and Poland

: 10/14/2022



The Microsoft Threat Intelligence Center (MSTIC) has identified evidence of a novel ransomware campaign targeting organizations in the transportation and related logistics industries in Ukraine and Poland utilizing a previously unidentified ransomware payload. We observed this new ransomware, which labels itself in its ransom note as “Prestige ranusomeware”, being deployed on October 11 in attacks occurring within an hour of each other across all victims.

This campaign had several notable features that differentiate it from other Microsoft-tracked ransomware campaigns:

- The enterprise-wide deployment of ransomware is not common in Ukraine, and this activity was not connected to any of the 94 currently active ransomware activity groups that Microsoft tracks
- The Prestige ransomware had not been observed by Microsoft prior to this deployment
- The activity shares victimology with recent Russian state-aligned activity, specifically on affected geographies and countries, and overlaps with previous victims of the FoxBlade malware (also known as HermeticWiper)

Despite using similar deployment techniques, the campaign is distinct from recent destructive attacks leveraging AprilAxe (ArguePatch)/CaddyWiper or Foxblade (HermeticWiper) that have impacted multiple critical infrastructure organizations in Ukraine over the last two weeks. MSTIC has not yet linked this ransomware campaign to a known threat group and is continuing investigations. MSTIC is tracking this activity as DEV-0960.

Microsoft uses DEV-#### designations as a temporary name given to an unknown, emerging, or a developing cluster of threat activity, allowing 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 or merged with existing actors.

This blog aims to provide awareness and indicators of compromise (IOCs) to Microsoft customers and the larger security community. Microsoft continues to monitor this and is in the process of early notification to customers impacted by DEV-0960 but not yet ransomed. MSTIC is also actively working with the broader security community and other strategic partners to share information that can help address this evolving threat through multiple channels.

Observed actor activity

Pre-ransomware activities

Prior to deploying ransomware, the DEV-0960 activity included the use of the following two remote execution utilities:

- RemoteExec – a commercially available tool for agentless remote code execution
- Impacket WMIexec – an open-source script-based solution for remote code execution

To gain access to highly privileged credentials, in some of the environments, DEV-0960 used these tools for privilege escalation and credential extraction:

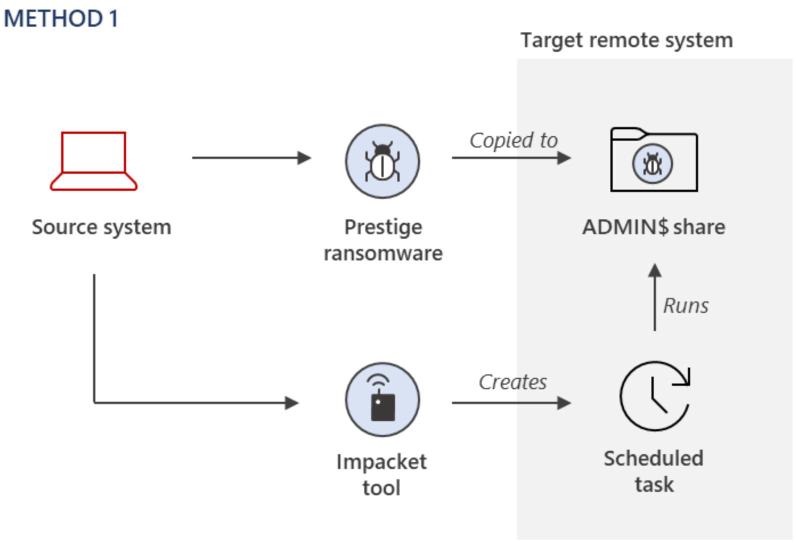
- winPEAS – an open-source collection of scripts to perform privilege escalation on Windows
- comsvcs.dll – used to dump the memory of the LSASS process and steal credentials
- ntdsutil.exe – used to back up the Active Directory database, likely for later use credentials

Ransomware deployment

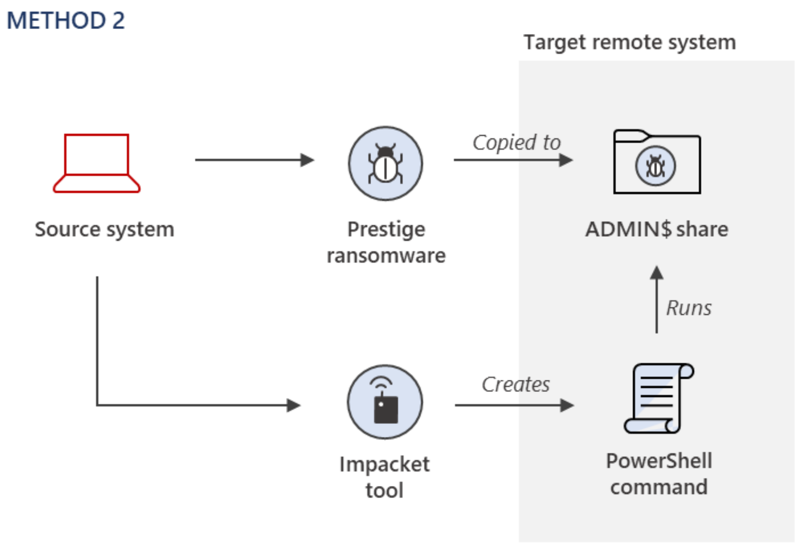
In all observed deployments, the attacker had already gained access to highly privileged credentials, like Domain Admin, to facilitate the ransomware deployment. Initial access vector has not been identified at this time, but in some instances it's possible that the attacker might have already had existing access to the highly privileged credentials from a prior compromise. In these instances, the attack timeline starts with the attacker already having Domain Admin-level access and staging their ransomware payload.

Most ransomware operators develop a preferred set of tradecraft for their payload deployment and execution, and this tradecraft tends to be consistent across victims, unless a security configuration prevents their preferred method. For this DEV-0960 activity, the methods used to deploy the ransomware varied across the victim environments, but it does not appear to be due to security configurations preventing the attacker from using the same techniques. This is especially notable as the ransomware deployments all occurred within one hour. The distinct methods for ransomware deployment were:

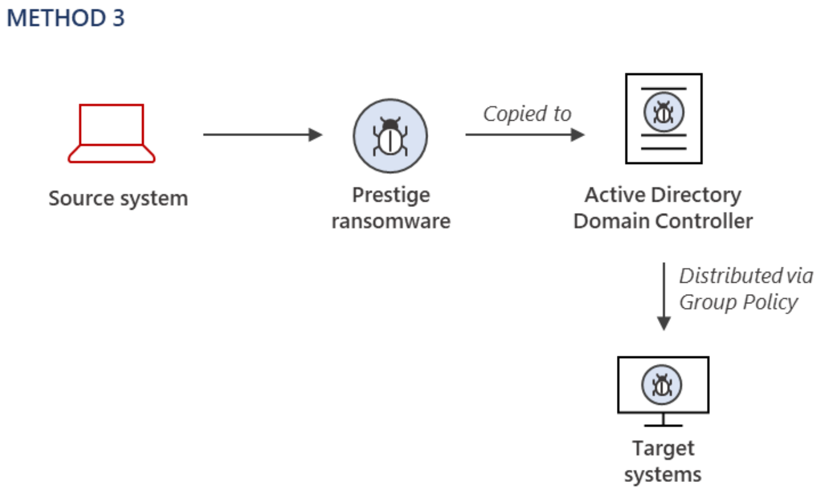
- **Method 1:** The ransomware payload is copied to the ADMIN$ share of a remote system, and Impacket is used to remotely create a Windows Scheduled Task on target systems to execute the payload



- **Method 2:** The ransomware payload is copied to the ADMIN$ share of a remote system, and Impacket is used to remotely invoke an encoded PowerShell command on target systems to execute the payload



- **Method 3:** The ransomware payload is copied to an Active Directory Domain Controller and deployed to systems using the Default Domain Group Policy Object

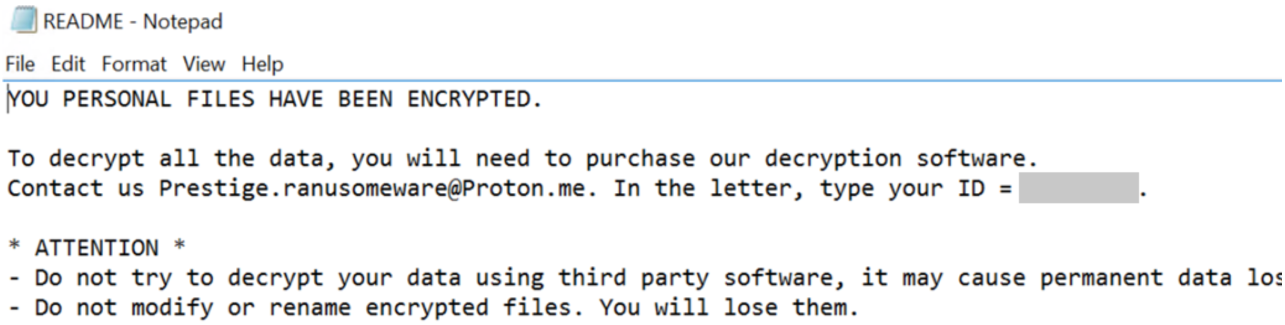


Malware analysis

The "Prestige" ransomware requires administrative privileges to run. Like many ransomware payloads, it attempts to stop the MSSQL Windows service to ensure successful encryption using the following command (the strings *"C:\Windows\System32\net.exe stop"* and *"MSSQLSERVER"* are both hardcoded in the analyzed samples):

```
C:\Windows\System32\net.exe stop MSSQLSERVER
```

Prestige creates *C:\Users\Public\README* and stores the following ransom note in the file. The same file is also created in the root directory of each drive:



```
YOU PERSONAL FILES HAVE BEEN ENCRYPTED.

To decrypt all the data, you will need to purchase our decryption software.
Contact us Prestige.ranusomeware@Proton.me. In the letter, type your ID = [redacted].

* ATTENTION *
- Do not try to decrypt your data using third party software, it may cause permanent data los
- Do not modify or rename encrypted files. You will lose them.
```

Prestige ransom note

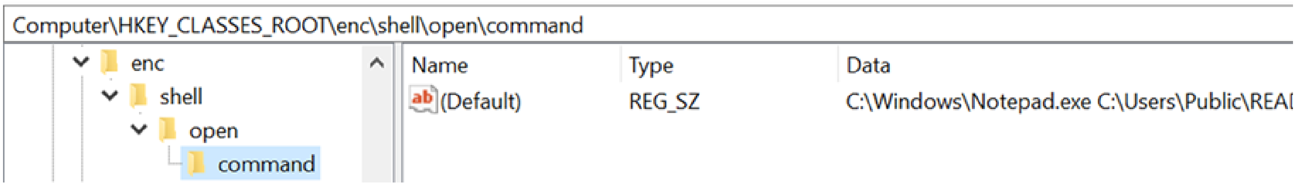
Prestige then traverses the files on the file system and encrypts the contents of files that have one of the following hardcoded file extensions, avoiding encrypting files in the *C:\Windows\* and *C:\ProgramData\Microsoft\* directories:

```
.1cd, .7z, .abk, .accdb, .accdc, .accde, .accdr, .alz, .apk, .apng, .arc, .asd,
.asf, .asm, .asx, .avhd, .avi, .avif, .bac, .backup, .bak, .bak2, .bak3, .bh, .bkp,
.bkup, .bkz, .bmp, .btr, .bz, .bz2, .bzip, .bzip2, .c, .cab, .cer, .cf, .cfu, .cpp,
.crt, .css, .db, .db-wal, .db3, .dbf, .der, .dmg, .dmp, .doc, .docm, .docx, .dot,
.dotm, .dotx, .dpx, .dsk, .dt, .dump, .dz, .ecf, .edb, .epf, .exb, .ged, .gif,
.gpg, .gzi, .gzip, .hdd, .img, .iso, .jar, .java, .jpeg, .jpg, .js, .json, .kdb,
.key, .lz, .lz4, .lzh, .lzma, .mdmr, .mkv, .mov, .mp3, .mp4, .mpeg, .myd, .nude,
.nvram, .oab, .odf, .ods, .old, .ott, .ovf, .p12, .pac, .pdf, .pem, .pfl, .pfx,
.php, .pkg, .png, .pot, .potm, .potx, .pps, .ppsm, .ppsx, .ppt, .pptm, .pptx, .prf,
.pvm, .py, .qcow, .qcow2, .r0, .rar, .raw, .rz, .s7z, .sdb, .sdc, .sdd, .sdf, .sfx,
.skey, .sldm, .sldx, .sql, .sqlite, .svd, .svg, .tar, .taz, .tbz, .tbz2, .tg, .tib,
.tiff, .trn, .txt, .txz, .tz, .vb, .vbox, .vbox-old, .vbox-prev, .vdi, .vdx, .vhd,
.vhdx, .vmc, .vmdk, .vmem, .vmsd, .vmsn, .vmss, .vmx, .vmxf, .vsd, .vsdx, .vss,
.vst, .vsx, .vtx, .wav, .wbk, .webp, .wmdb, .wmv, .xar, .xlm, .xls, .xlsb, .xlsm,
.xlsx, .xlt, .xltm, .xltx, .xlw, .xz, .z, .zbf, .zip, .zipx, .zl, .zpi, .zz
```

After encrypting each file, the ransomware appends the extension *.enc* to the existing extension of the file. For example, *changes.txt* is encrypted and then renamed to *changes.txt.enc*. Prestige uses the following two commands to register a custom file extension handler for files with *.enc* file extension:

```
C:\Windows\System32\reg.exe add HKCR\.enc /ve /t REG_SZ /d enc /f

C:\Windows\System32\reg.exe add HKCR\enc\shell\open\command /ve /t REG_SZ /d "C:\Windows\Notepad.exe
C:\Users\Public\README" /f
```



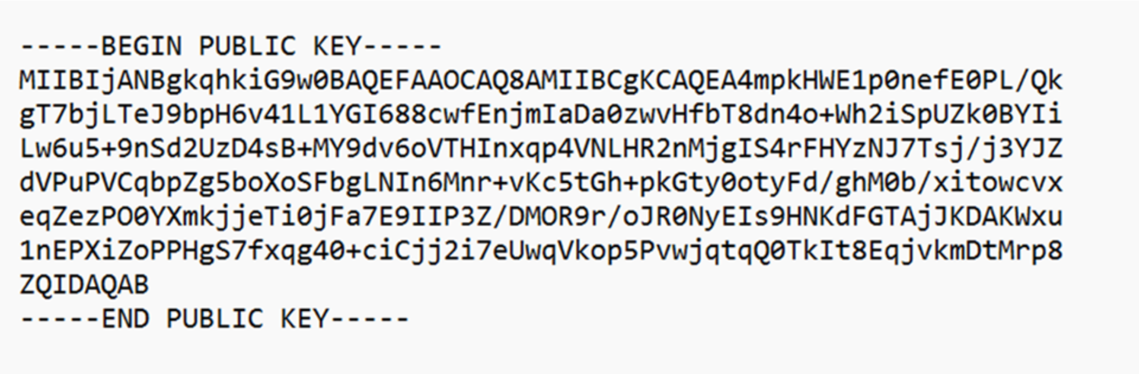
Computer\HKEY_CLASSES_ROOT\enc\shell\open\command

Name	Type	Data
(Default)	REG_SZ	C:\Windows\Notepad.exe C:\Users\Public\REAI

Custom file extension handler for files with *.enc* extension

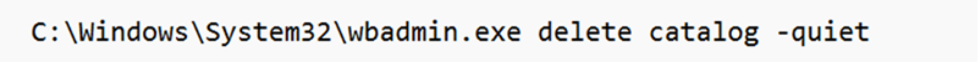
As a result of creating the custom file extension handler, when any file carrying the file extension *.enc* (i.e., encrypted by Prestige) is opened by a user, the file extension handler uses Notepad to open *C:\Users\Public\README*, which contains the ransom note.

To encrypt files, Prestige leverages the CryptoPP C++ library to AES-encrypt each eligible file. During the encryption process, the following hardcoded RSA X509 public key is used by one version of the ransomware (each version of Prestige may carry a unique public key):



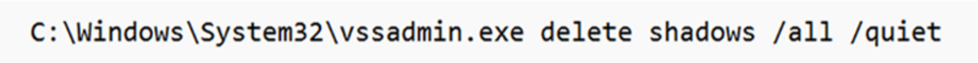
```
-----BEGIN PUBLIC KEY-----
MIIBIjANBgkqhkiG9w0BAQEFAAOCAQ8AMIIBCgKCAQEA4mpkHWE1p0nefE0PL/Qk
gT7bjLTeJ9bpH6v41L1YGI688cwfEnjmIaDa0zwvHfbT8dn4o+Wh2iSpUZk0BYIi
Lw6u5+9nSd2UzD4sB+MY9dv6oVTHInxqp4VNLHR2nMjgIS4rFHYzNJ7Tsj/j3YJZ
dVPuPVCqbpZg5boXoSFbgLNIn6Mnr+vKc5tGh+pkGty0otyFd/ghM0b/xitowcvx
eqZezPO0YXmkjjeTi0jFa7E9IIP3Z/DMOR9r/oJR0NyEIs9HNKdFGTAjJKDAKWxu
1nEPXiZoPPHgS7fxqg40+ciCjj2i7eUwqVkop5PvwjqtqQ0TkIt8EqjvkmDtMrp8
ZQIDAQAB
-----END PUBLIC KEY-----
```

To hinder system and file recovery, Prestige runs the following command to delete the backup catalog from the system:



```
C:\Windows\System32\wbadmin.exe delete catalog -quiet
```

Prestige also runs the following command to delete all volume shadow copies on the system:



```
C:\Windows\System32\vssadmin.exe delete shadows /all /quiet
```

Before running the commands above, the 32-bit version of Prestige calls the function Wow64DisableWow64FsRedirection() to disable file system redirection and gain access to the native System32 directory. After running the commands above, Prestige restores file system redirection by calling the function Wow64RevertWow64FsRedirection().

Microsoft will continue to monitor DEV-0960 activity and implement protections for our customers. The current detections, advanced detections, and IOCs in place across our security products are detailed below.

Looking forward

The threat landscape in Ukraine continues to evolve, and wipers and destructive attacks have been a consistent theme. Ransomware and wiper attacks rely on many of the same security weaknesses to succeed. As the situation evolves, organizations can adopt the hardening guidance below to help build more robust defenses against these threats.

Recommended customer actions

The ransomware payload was deployed by the actor after an initial compromise that involved gaining access to highly privileged credentials. The techniques used by the actor and described in the “Observed Actor Activity” section can be mitigated by adopting the security considerations provided below:

- Block process creations originating from PSExec and WMI commands to stop lateral movement utilizing the WMIexec component of Impacket.
- Enable Tamper protection to prevent attacks from stopping or interfering with Microsoft Defender.
- Turn on cloud-delivered protection in Microsoft Defender Antivirus or the equivalent for your antivirus product to cover rapidly evolving attacker tools and techniques. Cloud-based machine learning protections block a huge majority of new and unknown variants.
- While this attack differs from traditional ransomware, following our defending against ransomware guidance helps protect against the credential theft, lateral movement, and ransomware deployment used by DEV-0960.
- Use the included indicators of compromise to investigate whether they exist in your environment and assess for potential intrusion.
- Enable multifactor authentication (MFA) to mitigate potentially compromised credentials and ensure that MFA is enforced for all remote connectivity, including VPNs. Microsoft strongly encourages all customers download and use password-less solutions like Microsoft Authenticator to secure your accounts.

Indicators of compromise (IOCs)

The following table lists the IOCs observed during our investigation. We encourage our customers to investigate these indicators in their environments and implement detections and protections to identify past related activity and prevent future attacks against their systems.

Indicator	Type	Description
5dd1ca0d471dee41eb3ea0b6ea117810f228354fc3b7b47400a812573d40d91d	SHA-256	Prestige ransomware payload
5fc44c7342b84f50f24758e39c8848b2f0991e8817ef5465844f5f2ff6085a57	SHA-256	Prestige ransomware payload
6cff0bbd62efe99f381e5cc0c4182b0fb7a9a34e4be9ce68ee6b0d0ea3eee39c	SHA-256	Prestige ransomware payload
a32bbc5df4195de63ea06feb46cd6b55	Import hash	Unique PE Import Hash shared by ransomware payloads

C:\Users\Public\README	File path	File path of the ransom note

NOTE: These indicators should not be considered exhaustive for this observed activity.

Detections

Microsoft 365 Defender

Microsoft Defender Antivirus

Microsoft Defender Antivirus detects known Prestige ransomware payloads with the following detection:

- Ransom:Win32/Prestige

Microsoft Defender for Endpoint

Microsoft Defender for Endpoint provides alerts for the indicators used by DEV-0960 discussed above.

- Ransomware-linked emerging threat activity group DEV-0960 detected

Microsoft Defender for Endpoint also provides alerts for the pre-ransom techniques discussed above.

Customers should act on these alerts as they indicate hands-on-keyboard attacks. **NOTE:** These alerts are not uniquely tied to the Prestige ransomware nor to the campaign discussed.

- Ongoing hands-on-keyboard attack via Impacket toolkit
- WinPEAS tool detected
- Sensitive credential memory read
- Password hashes dumped from LSASS memory
- Suspicious scheduled task activity
- System recovery setting tampering
- File backups were deleted

Advanced hunting queries

Microsoft Sentinel

Prestige ransomware file hashes

This query looks for file hashes and Microsoft Defender Antivirus detections associated with Prestige ransomware payload.

- https://github.com/Azure/Azure-Sentinel/tree/master/Detections/MultipleDataSources/PrestigeRansomwareIOCsOct22.yaml

Microsoft 365 Defender

Impacket WMIexec usage

This query surfaces Impacket WMIexec usage on a device:

```
DeviceProcessEvents
| where Timestamp >= ago(7d)
| where FileName =~ "cmd.exe"
| where ProcessCommandLine has_all (@" 1> \127.0.0.1\", "/Q ", "/c ", @" 2>&1")
| where InitiatingProcessFileName =~ "WmiPrvSE.exe"
```

This query has the same purpose as above, but it also groups all the commands launched using Impacket WMIexec on the device:

```
DeviceProcessEvents
| where Timestamp >= ago(7d)
| where FileName =~ "cmd.exe"
| where ProcessCommandLine has_all (@" 1> \127.0.0.1\", "/Q ", "/c ", @" 2>&1")
| where InitiatingProcessFileName =~ "WmiPrvSE.exe"
| project DeviceName, DeviceId, Timestamp, ProcessCommandLine
| summarize make_set(ProcessCommandLine), min(Timestamp), max(Timestamp) by DeviceId,
DeviceName
```

LSASS process memory dumping

This query surfaces attempts to dump the LSASS process memory comsvcs.dll:

```
let startTime = ago(7d);
let endTime = now();
DeviceProcessEvents
| where Timestamp between (startTime..endTime)
| where FileName =~ 'rundll32.exe'
and ProcessCommandLine has 'comsvcs.dll'
and ProcessCommandLine has_any ('full','MiniDump')
| where not (ProcessCommandLine matches regex @'{[\w\d]{8}-[\w\d]{4}-[\w\d]{4}-[\w\d]
{4}-[\w\d]{12}}'
and ProcessCommandLine matches regex @'(\d{2}_){3}' )
```