

MAR-10265965-1.v1 – North Korean Trojan: BISTROMATH | CISA

Published: 2020-02-14 · Archived: 2026-04-05 16:51:10 UTC

Notification

This report is provided "as is" for informational purposes only. The Department of Homeland Security (DHS) does not provide any warranties of any kind regarding any information contained herein. The DHS does not endorse any commercial product or service referenced in this bulletin or otherwise.

This document is marked TLP:WHITE--Disclosure is not limited. Sources may use TLP:WHITE when information carries minimal or no foreseeable risk of misuse, in accordance with applicable rules and procedures for public release. Subject to standard copyright rules, TLP:WHITE information may be distributed without restriction. For more information on the Traffic Light Protocol (TLP), see <http://www.us-cert.gov/tlp>.

Summary

Description

This Malware Analysis Report (MAR) is the result of analytic efforts between Department of Homeland Security (DHS), the Federal Bureau of Investigation (FBI), and the Department of Defense (DoD). Working with U.S. Government partners, DHS, FBI, and DoD identified Trojan malware variants used by the North Korean government. This malware variant has been identified as BISTROMATH. The U.S. Government refers to malicious cyber activity by the North Korean government as HIDDEN COBRA. For more information on HIDDEN COBRA activity, visit <https://www.us-cert.gov/hiddencobra>.

DHS, FBI, and DoD are distributing this MAR to enable network defense and reduce exposure to North Korean government malicious cyber activity.

This MAR includes malware descriptions related to HIDDEN COBRA, suggested response actions and recommended mitigation techniques. Users or administrators should flag activity associated with the malware and report the activity to the Cybersecurity and Infrastructure Security Agency (CISA) or the FBI Cyber Watch (CyWatch), and give the activity the highest priority for enhanced mitigation.

This report looks at multiple versions of a full-featured RAT implant executable and multiple versions of the CAgent11 GUI implant controller/builder. These samples performs simple XOR network encoding and are capable of many features including conducting system surveys, file upload/download, process and command execution, and monitoring the microphone, clipboard, and the screen. The GUI controllers allow interaction with the implant as well as the option to dynamically build new implants with customized options. The implants are loaded with a trojanized executable containing a fake bitmap which decodes into shellcode which loads the embedded implant.

For a downloadable copy of IOCs, see [MAR-101265965-1.v1.stix](#).

Submitted Files (5)

04d70bb249206a006f83db39bbe49ff6e520ea329e5fbb9c758d426b1c8dec30 (96071956D4890AEBEA14ECD8015617...)
1ea6b3e99bbb67719c56ad07f5a12501855068a4a866f92db8dcdefaffa48a39 (688890DDBF532A4DE7C83A58E6AA59...)
618a67048d0a9217317c1d1790ad5f6b044eaa58a433bd46ec2fb9f9ff563dc6 (0AE8A7B6B4D70C0884095629FC02C1...)
738ba44188a93de6b5ca7e0bf0a77f66f677a0dda2b2e9ef4b91b1c8257da790 (C51416635E529183CA5337FADE8275...)
b6811b42023524e691b517d19d0321f890f91f35ebddf1c12cbb92cda5b6de32 (26520499A3FC627D335E34586E99DE...)

Additional Files (2)

133820ebac6e005737d5bb97a5db549490a9f210f4e95098bc9b0a7748f52d1f (a21171923ec09b9569f2baad496c9e...)
43193c4efa8689ff6de3fb18e30607bb941b43abb21e8cee0cfd664c6f4ad97c (83833f8bdd6ecf3a1212f5d1fc3d9...)

IPs (1)

159.100.250.231

Findings

1ea6b3e99bbb67719c56ad07f5a12501855068a4a866f92db8dcdefaffa48a39

Tags

backdooremotetrojan

Details

Name	688890DDBF532A4DE7C83A58E6AA594F
Name	ss.exe
Size	1102926 bytes
Type	PE32 executable (GUI) Intel 80386, for MS Windows
MD5	688890ddb532a4de7c83a58e6aa594f
SHA1	d8f6a7f32c929ce9458691447ff1cf6d180588c8
SHA256	1ea6b3e99bbb67719c56ad07f5a12501855068a4a866f92db8dcdefaffa48a39
SHA512	8484bea6adf27c2323632c3e94f91eb313e341622b5696b0d24105be1f24fa356f5fceb8fc691e2d309fd24f7d8bb41fd7b682c29193128a
ssdeep	24576:kgWxnOH3vvS+7nD03glQ1J6cS2lvyip5HkRpB7T4IRMh3y:kgWZMvSKnY3DJLSoORT7ThAC
Entropy	7.951069

Antivirus

Ahnlab	Trojan/Win32.Bmdoor
Antiy	Trojan[Backdoor]/Win32.Androm
Avira	TR/Injector.ukfuc
BitDefender	Trojan.GenericKD.41987827
ClamAV	Win.Trojan.Agent-7376538-0
Cyren	W32/Trojan.IZTF-2035
ESET	a variant of Win32/Injector.DQTY trojan
Emsisoft	Trojan.GenericKD.41987827 (B)
Ikarus	Trojan.Win32.Injector
K7	Riskware (0040eff71)
McAfee	Trojan-Injector.c
Microsoft Security Essentials	Trojan:Win32/Agentesla!MTB
NANOAV	Trojan.Win32.Androm.ghyuau
Sophos	Troj/Inject-ETF
Symantec	Backdoor.Tidserv
Systweak	trojan.injector
TACHYON	Backdoor/W32.Androm.1102926
TrendMicro	TROJ_FR.7170E263
TrendMicro House Call	TROJ_FR.7170E263
VirusBlokAda	Backdoor.Androm
Zillya!	Backdoor.Androm.Win32.44606

YARA Rules

```

• rule CryptographyFunction
{
  meta:
    author = "CISA trusted 3rd party"
    incident = "10271944.r1.v1"
    date = "2019-12-25"
    category = "Hidden_Cobra"
    family = "HOTCROISSANT"
  strings:
    $ALGO_crypto_1 = { 8A [1-5] 32 [1-4] 32 [1-4] 32 [1-4] 88 [1-5] 8A [1-4] 32 [1-4] 22 [1-4] 8B [1-5] 8D [3-7]
33 [1-4] 81 [3-7] C1 [1-5] C1 [1-5] 0B [1-4] 8D [1-5] 33 [1-4] 22 [1-4] C1 [1-5] 33 [1-4] 32 [1-4] 8B [1-4] 83 [1-5]
C1 [1-5] 33 [1-4] C1 [1-5] C1 }
    condition:
      uint16(0) == 0x5A4D and any of them
}

```

ssdeep Matches

No matches found.

PE Metadata

Compile Date	2008-01-17 10:34:19-05:00
Import Hash	68d3c5fd0c41042f190fa12a4eebfe1b

PE Sections

MD5	Name	Raw Size	Entropy
0b8ab9af886c4161371944bd46af685d	header	1024	2.484025
0cc984b88cda683bad52d886fbadf22d	.text	77824	6.585222
d7200a9095f81e46d89eb2175a7d16ba	.rdata	21504	4.940483
56eae295cdc645a889cc51643c19ca1c	.data	5632	3.200450
31d4e62663767a64bd72b957df2bed2e	.rsrc	1536	4.029623
c7a9818fe1b1f64be18f67db25dbed6d	.reloc	7680	4.982554

Packers/Compilers/Cryptors

Relationships

1ea6b3e99b...	Connected_To	159.100.250.231
1ea6b3e99b...	Contains	43193c4efa8689ff6de3fb18e30607bb941b43abb21e8cee0cfd664c6f4ad97c

Description

The samples use a PlanetCPP.com 'RichEdit example' executable to obfuscate calling a decryption function which decrypts an embedded 'fake' bitmap image into the configuration and shellcode. When the malicious function is called, it deobfuscates API pointers, loads the full file into memory, calculates an offset into the memory to a 'fake' bitmap image, decodes the image; which becomes configuration options and shellcode and then executes the shellcode.

The embedded shellcode has many selectable options.

```

-----Begin Shellcode Options-----
- option00: Embedded vs Downloaded payload
0 -> payload embedded within own file at offset (option27 + option28 + option22)
1 -> Download payload from url <option30> to %temp%\<option31>\RGID3D88.tmp

```

- option01: True -> check for vm artifacts:

registry checks:

- VMWARE Scsi device
- VBOX Scsi device
- QEMU Scsi device
- SOFTWARE\Vmware,Inc.\Vmware_Tools
- HARDWARE\Description\System\SystemBiosVersion == "VBOX"
- HARDWARE\Description\System\SystemBiosVersion == "QEMU"
- HARDWARE\Description\System\SystemBiosVersion == "BOCHS"
- HARDWARE\Description\System\VideoBiosVersion == "VIRTUALBOX"
- HARDWARE\Description\System\SystemBiosDate == 06/23/99
- SOFTWARE\Oracle\VirtualBox_Guest_Additions
- HARDWARE\ACPI\DSDT\VBOX_
- HARDWARE\ACPI\FADT\VBOX_
- HARDWARE\ACPI\RSDT\VBOX_
- SYSTEM\ControlSet001\Services\VBoxGuest
- SYSTEM\ControlSet001\Services\VBoxMouse
- SYSTEM\ControlSet001\Services\VBoxService
- SYSTEM\ControlSet001\Services\VBoxSF
- SYSTEM\ControlSet001\Services\VBoxVideo

file checks:

- C:\WINDOWS\system32\drivers\vmmouse.sys
- C:\WINDOWS\system32\drivers\vmhgfs.sys
- \\.\HGFS
- \\.\vmci
- C:\WINDOWS\system32\drivers\VBoxMouse.sys
- C:\WINDOWS\system32\drivers\VBoxGuest.sys
- C:\WINDOWS\system32\drivers\VBoxSF.sys
- C:\WINDOWS\system32\drivers\VBoxVideo.sys
- C:\WINDOWS\system32\vboxdisp.dll
- C:\WINDOWS\system32\vboxhook.dll
- C:\WINDOWS\system32\vboxmrxnp.dll
- C:\WINDOWS\system32\vboxogl.dll
- C:\WINDOWS\system32\vboxoglarrayspu.dll
- C:\WINDOWS\system32\vboxoglcrutil.dll
- C:\WINDOWS\system32\vboxoglerrorspu.dll
- C:\WINDOWS\system32\vboxoglfeedbackspu.dll
- C:\WINDOWS\system32\vboxoglpacspu.dll
- C:\WINDOWS\system32\vboxoglpassthroughspu.dll
- C:\WINDOWS\system32\vboxservice.exe
- C:\WINDOWS\system32\vboxtray.exe
- C:\WINDOWS\system32\VBoxControl.exe
- C:\program_files\oracle\virtualbox_guest_additions
- \\.\VBoxMiniRdrDN
- \\.\pipe\VBoxMiniRdDN
- \\.\VBoxTrayIPC
- \\.\pipe\VBoxTrayIPC

Network Adapter checks:

- Check for Vmware MAC addresses
- Check for VirtualBox MAC addresses
- Check for VMware network adapter

Window Checks:

- VBoxTrayToolWndClass
- VBoxTrayToolWnd

Process Checks:

- vboxservice.exe
- vboxtray.exe

Loaded DLLs:

- vmcheck.dll

- option02: True -> check for sandbox artifacts:
 - Verify spin loops aren't skipped
 - Verify kernel32 doesn't contain export "wine_get_unix_file_name"
 - Verify Numa api calls are not bypassed
 - Loaded DLLs:
 - SbieDll.dll
 - api_log.dll
 - dir_watch.dll
 - dbghelp.dll
 - wpespy.dll
 - registry checks:
 - SOFTWARE\Wine
 - file checks:
 - C:\sandbox\sandbox.exe
 - C:\sandbox\sbfwe.dll
 - username checks:
 - SANDBOX
 - VIRUS
 - MALWARE
 - SCHMIDTI
 - CURRENTUSER
 - ANDY
 - current directory checks:
 - VIRUS
 - SANDBOX
 - SAMPLE
- option03: True -> check for debugging artifacts:
 - API calls:
 - IsDebuggerPresent
 - CheckRemoteDebuggerPresent
 - NtQueryInformationProcess
 - GetThreadContext
 - OutputDebugString
- option04: Check if certain processes are running:
 - 0 -> ignored
 - 1 -> exit if specific processes are running
 - 2 -> exit if specific processes are not running
 - parses option31_array_+0x200 for a list of ,;: separated process names
- option05: Queries Software\Microsoft\Windows\CurrentVersion\Uninstall keys
 - exits if return value is != 0
- option06: Check for specific languages
 - 0 -> ignored
 - 1 -> exit if current language is found in list
 - 2 -> exit if current language is not found in list
 - parses option31_array_+0x4b0 for a list of ,;: separated languages
- option07: Check for specific usernames
 - 0 -> ignored
 - 1 -> exit if current username is found in list
 - 2 -> exit if current username is not found in list
 - parses option31_array_+0x6b8 for a list of ,;: separated usernames
- option08: Check for specific computernames
 - 0 -> ignored
 - 1 -> exit if current computernames is found in list
 - 2 -> exit if current computernames is not found in list
 - parses option31_array_+0x8ac for a list of ,;: separated computernames

- option09: Something with querying Software\Microsoft\Windows\CurrentVersion\Uninstall keys
exits if return value is < option09_value
- option10: integer value -> exits if there are fewer than this many processes running
- option11-14: Check for system/drive info
 - 11==0x001 -> exit if number of processors <= option12
 - 11==0x010 -> exit if total physical memory <= option13
 - 11==0x100 -> exit if total harddisk space <= option14
- option12/27/28: if True -> exploit dll hijack in cliconfg.exe (SQL Server Client Network Utility)
dumps a number (option28) of bytes from an offset (option27) of this file into %temp%\ntwdblib.dll
creates a Software\Claiomh registry key
executes cliconfg.exe (which loads ntwdblib.dll)
- option16: Set EnableLUA registry key
SOFTWARE\Microsoft\Windows\CurrentVersion\Policies\System\EnableLUA to <option16>
- option17: Create Persistence
 - 0 -> ignored
 - 1 -> Add registry key to Software\Microsoft\Windows\CurrentVersion\Run using a name from option31_array_+0x960
 - 2 -> Copy self into Startup folder
 - 3 -> Create an hourly Scheduled Task called "System Backup"
- option18/23: Process Hollowing vs Drop/Execute
 - == 0 -> Do Process Hollowing
 - != 0 -> Dump payload to file and execute directly:
 - write to %temp%\RT5380.exe using own file offset (option27 + option28 + option22) and execute
 - write to %temp%\<option30> using own file offset (option27 + option28 + option22) and execute
 - check option23:
 - ==0 -> ignored
 - !=0 -> delete self and replace self with the dropped file
- option19: Process to create/hollow/inject/execute
 - 0 -> self
 - 1 -> svchost.exe
 - 2 -> conhost.exe
 - 3 -> explorer.exe
 - 4 -> value of "http\shell\open\command" registry key
 - 5 -> <option33>
- option20: Sleep timer
Milliseconds to sleep before doing process hollowing
- option21/26: Kill timer
 - 0 -> ignored
 - 1 -> if timestamp of module + <option26> >= currentTime -> remove persistence, delete self, exit process
- option29/34/35: move file to desired location, delete old file, and execute from new location
 - additional path is in option34
 - new filename is in option35
 - 0 -> C:\
 - 1 -> %windir%
 - 2 -> %system%
 - 3 -> %programfiles%
 - 4 -> %programfiles%\Common Files\
 - 5 -> C:\ProgramData\
 - 6 -> %userprofile%
 - 7 -> %userprofile%\Documents\
 - 8 -> %temp%
 - 9 -> %userprofile%\Favorites\
 - 10 -> %appdata%\n
 - 11 -> %localappdata%

- option36: char[40] - Unknown - Possibly adds a mutex to the hollowed process to enforce a single execution
 - Uses argument to create a named mutex
 - Injects additional code into the hollowed process (from offset 0x28c0)
 - Injects <option36> into the hollowed process
 - Creates another remote thread in the hollowed process pointing at offset 0x465a of the newly injected memory
- End Shellcode Options-----

Screenshots

Figure 1: Implant Functionality -

618a67048d0a9217317c1d1790ad5f6b044eaa58a433bd46ec2fb9f9ff563dc6

Tags

dropperremotetkeyloggerspywaretrojan

Details

Name	0AE8A7B6B4D70C0884095629FC02C19C
Name	CAgent11.exe
Size	13498368 bytes
Type	PE32 executable (GUI) Intel 80386, for MS Windows
MD5	0ae8a7b6b4d70c0884095629fc02c19c
SHA1	9efa2d68932ff24cb18eb7e35aa5f91ce99596e8
SHA256	618a67048d0a9217317c1d1790ad5f6b044eaa58a433bd46ec2fb9f9ff563dc6
SHA512	08f724812cbeff4020ac3fb07cafec5cde17f53f4644d554351cf4056907a6363d5b21ed3720976820307b43a543e81c6cc27c241f4449fd9
ssdeep	196608:Klq/1ui17DaLU114O5dm/+f99FLOyomFHKnPG:GcvlMLMg/299F
Entropy	5.658332

Antivirus

Ahnlab	Dropper/Win32.Keylogger
Antiy	Trojan[Spy]/Win32.Agent
Avira	HEUR/AGEN.1038092
Cyren	W32/Agent.RBBJ-4429
ESET	a variant of Win32/Spy.Agent.PUH trojan
Ikarus	Trojan-Spy.Agent
K7	Spyware (00555d821)
McAfee	Trojan-Injector.d
Microsoft Security Essentials	Trojan:Win32/Emotet
NANOAV	Trojan.Win32.Graftor.ggzicq
NetGate	Trojan.Win32.Malware
Sophos	Troj/Agent-BCXS
Symantec	Trojan Horse
Systweak	malware.keylogger
TACHYON	Trojan/W32.Keylogger.13498368

VirusBlokAda	TrojanSpy.Agent
Zillya!	Trojan.Agent.Win32.1169060

YARA Rules

No matches found.

ssdeep Matches

No matches found.

PE Metadata

Compile Date	2017-03-21 21:12:17-04:00
Import Hash	c4406c66f7ca84ffb881d843c49acbd6

PE Sections

MD5	Name	Raw Size	Entropy
e7e02cd4a189cea5efaa8fb36509aa45	header	1024	3.530105
d41d8cd98f00b204e9800998ecf8427e	.textbss	0	0.000000
5db50cefb12a73d10aad429548befe7	.text	7047680	5.565086
e9a63040b7f3e75b5746d8202d8594f5	.rdata	904704	4.415613
1e815bbe0c5cadf4953bbaac6259dcaa	.data	40448	4.299279
16342b710a408579ee34f3ccf9927331	.idata	28672	5.161732
c573bd7cea296a9c5d230ca6b5aee1a6	.tls	1024	0.011174
011d6c8672f924dc710a68acb6bc74f9	.00cfg	512	0.061163
867de3faa85f377519582ed29a83384c	.rsrc	5123072	4.951562
e74f13482e13eb316d544b69a046ff15	.reloc	351232	6.011950

Packers/Compilers/Cryptors

Description

See analysis for "04D70BB249206A006F83DB39BBE49FF6E520EA329E5FBB9C758D426B1C8DEC30".

Implants built with sample "04D70BB249206A006F83DB39BBE49FF6E520EA329E5FBB9C758D426B1C8DEC30" are not compatible with this controller, and vice versa.

b6811b42023524e691b517d19d0321f890f91f35ebbf1c12cbb92cda5b6de32

Tags

backdooremotetrojan

Details

Name	26520499A3FC627D335E34586E99DE7A
Name	ADManager.exe
Size	1120318 bytes
Type	PE32 executable (GUI) Intel 80386, for MS Windows
MD5	26520499a3fc627d335e34586e99de7a

SHA1	df10c097e42dbe7ea4478a984c5e2ab586147519
SHA256	b6811b42023524e691b517d19d0321f890f91f35ebddf1c12cbb92cda5b6de32
SHA512	898ab1a1cd5a731e94a7b4c0a274e81092fe6de2ea888b3db2d22cf4d0bacbbb36f486152ff10f61f054091aee421f00d89a8741fce0f370cc
ssdeep	24576:3gWPfTO4H59Z6PTvnh2gf2JfvoioZ74XKBpNCY+SOToKMcxGa52w:3gW3S4Z9ATcggox4wpwYq9Mcx3B
Entropy	7.953591

Antivirus

Ahnlab	Backdoor/Win32.Androm
Antiy	Trojan[Backdoor]/Win32.Androm
Avira	TR/Injector.cskrn
BitDefender	Trojan.GenericKD.41987802
ClamAV	Win.Trojan.Agent-7376533-0
Cyren	W32/Androm.DKHG-0510
ESET	a variant of Win32/Injector.DQTY trojan
Emsisoft	Trojan.GenericKD.41987802 (B)
Ikarus	Trojan.Win32.Injector
K7	Riskware (0040eff71)
McAfee	Trojan-Injector.c
Microsoft Security Essentials	Trojan:Win32/Agentesla!MTB
NANOAV	Trojan.Win32.Androm.ggadbc
Sophos	Troj/Inject-ETF
Symantec	Trojan Horse
Systweak	trojan.injector
TACHYON	Backdoor/W32.Androm.1120318
TrendMicro	TROJ_FR.7170E263
TrendMicro House Call	TROJ_FR.7170E263
VirusBlokAda	Backdoor.Androm
Zillya!	Backdoor.Androm.Win32.44606

YARA Rules

- rule CryptographyFunction
 - {
 - meta:
 - author = "CISA trusted 3rd party"
 - incident = "10271944.r1.v1"
 - date = "2019-12-25"
 - category = "Hidden_Cobra"
 - family = "HOTCROISSANT"
 - strings:
 - \$ALGO_crypto_1 = { 8A [1-5] 32 [1-4] 32 [1-4] 32 [1-4] 88 [1-5] 8A [1-4] 32 [1-4] 22 [1-4] 8B [1-5] 8D [3-7] 33 [1-4] 81 [3-7] C1 [1-5] C1 [1-5] 0B [1-4] 8D [1-5] 33 [1-4] 22 [1-4] C1 [1-5] 33 [1-4] 32 [1-4] 8B [1-4] 83 [1-5] C1 [1-5] 33 [1-4] C1 [1-5] C1 }
 - condition:

```
uint16(0) == 0x5A4D and any of them
}
```

ssdeep Matches

No matches found.

PE Metadata

Compile Date	2017-03-26 09:21:10-04:00
Import Hash	68d3c5fd0c41042f190fa12a4eebfe1b

PE Sections

MD5	Name	Raw Size	Entropy
a507172c7e89d3f88c70c4fd6827a522	header	1024	2.476553
0cc984b88cda683bad52d886fbadf22d	.text	77824	6.585222
d7200a9095f81e46d89eb2175a7d16ba	.rdata	21504	4.940483
56eae295cdc645a889cc51643c19ca1c	.data	5632	3.200450
58dbdc33cb7f42b5e3a9f0fcc94d6b1f	.rsrc	1024	4.796047
c7a9818fe1b1f64be18f67db25dbed6d	.reloc	7680	4.982554

Packers/Compilers/Cryptors

Relationships

b6811b4202...	Connected_To	159.100.250.231
b6811b4202...	Contains	133820ebac6e005737d5bb97a5db549490a9f210f4e95098bc9b0a7748f52d1f

Description

See analysis for file "1ea6b3e99bbb67719c56ad07f5a12501855068a4a866f92db8cdefaffa48a39" for additional details.

738ba44188a93de6b5ca7e0bf0a77f66f677a0dda2b2e9ef4b91b1c8257da790

Tags

trojan

Details

Name	C51416635E529183CA5337FADE82758A
Name	server.exe
Size	947200 bytes
Type	PE32 executable (GUI) Intel 80386, for MS Windows
MD5	c51416635e529183ca5337fade82758a
SHA1	830368d88b661d09c084e484713effb8d230d328
SHA256	738ba44188a93de6b5ca7e0bf0a77f66f677a0dda2b2e9ef4b91b1c8257da790
SHA512	244b67e0b9e9ab2fa6ccceeb4ad71207f1d8371af9c69af93bcc15cc8b592aca54e9c241d439b94ed28923d4622050fccdc38b326a8d15b8:
ssdeep	24576:9oV9SPwODditnxk93QKTrCEgqAGYOEgJZ+0Mn:9o2I2du23QxErv7ESZ+7n
Entropy	6.703705

Antivirus

Ahnlab	Malware/Win32.Generic
Antiy	Trojan/Win32.AGeneric
Avira	HEUR/AGEN.1038092
BitDefender	Trojan.GenericKD.32683846
ClamAV	Win.Trojan.Agent-7376468-0
Cyren	W32/Agent.KUBI-8127
ESET	a variant of Win32/Agent.SSC trojan
Emsisoft	Trojan.GenericKD.32683846 (B)
Ikarus	Trojan.Win32.Agent
K7	Trojan (0027657e1)
McAfee	Generic Trojan.sh
NANOAV	Trojan.Win32.TrjGen.ghyubn
Sophos	Troj/Agent-BCXS
Symantec	Trojan Horse
Systweak	malware.passwordstealer
TrendMicro	TROJ_FR.7170E263
TrendMicro House Call	TROJ_FR.7170E263
VirusBlokAda	BScope.TrojanSpy.Agent
Zillya!	Trojan.Agent.Win32.1168332

YARA Rules

No matches found.

ssdeep Matches

No matches found.

PE Metadata

Compile Date	2017-04-13 23:44:03-04:00
Import Hash	d31e404296b957729148721e11f3bc88

PE Sections

MD5	Name	Raw Size	Entropy
1db5d7f5d8e2fa35f4077d3c28b60ae7	header	1024	3.229935
6f6469c660281de2c72fa3685d55a8ec	.text	710656	6.655052
0847400b5430782ad644a30cd8240c73	.rdata	167424	5.776485
77ab2f92d6177b9e39430447aa595073	.data	37376	5.315603
1f354d76203061bfd5a53dae48d5435	.tls	512	0.020393
1704ffd93e9d463dc42784bc03bbfd5d	.gfids	512	2.779799
850aa99c8c1a85dc7545811d66bb0c17	.rsrc	512	4.717679

MD5	Name	Raw Size	Entropy
48da542e50cc8e12bdb9cab38a8ce0cb	.reloc	29184	6.576636

Packers/Compilers/Cryptors

Relationships

738ba44188...	Connected_To	159.100.250.231
---------------	--------------	-----------------

Description

This sample is a full-featured RAT executable.

See analysis for file "1ea6b3e99bbb67719c56ad07f5a12501855068a4a866f92db8dcdefaffa48a39" for additional details.

This sample varies slightly in the following ways.

Victim_info for this version contains Unicode strings. The RAT is controllable by an unknown variant of CAgent.exe.

04d70bb249206a006f83db39bbe49ff6e520ea329e5fbb9c758d426b1c8dec30

Tags

dropperemotetkeyloggerspywaretrojan

Details

Name	96071956D4890AEBEA14ECD8015617CC
Name	CAgent11.exe
Size	7014400 bytes
Type	PE32 executable (GUI) Intel 80386, for MS Windows
MD5	96071956d4890aeb14ecd8015617cc
SHA1	49e16180795034a4888fff776968e29871f79340
SHA256	04d70bb249206a006f83db39bbe49ff6e520ea329e5fbb9c758d426b1c8dec30
SHA512	29abd5fa0c24e42916631f830b6860027dcefd320978bee389e55f4f04278668ec4cfb67e5b1c8b7133338cc0fb09ffae28c5cf6d5226d1f
ssdeep	98304:SC6l4uHxECiYwS2BsszjfisjJiBg1pDClmMFLOakGkzdnEVomFHKnP:P44uHi0mFi+1p+FLOYomFHKnP
Entropy	5.907837

Antivirus

Ahnlab	Dropper/Win32.Keylogger
Avira	HEUR/AGEN.1038092
BitDefender	Trojan.GenericKD.32683845
Cyren	W32/Trojan.KVTC-7019
ESET	a variant of Win32/Spy.Agent.PUH trojan
Emsisoft	Trojan.GenericKD.32683845 (B)
Ikarus	Trojan-Spy.Agent
K7	Spyware (00555d821)
McAfee	Trojan-Injector.d
Microsoft Security Essentials	Trojan:Win32/Emotet

NANOAV	Trojan.Win32.TrjGen.ghyuap
Sophos	Troj/Agent-BCXS
Symantec	Trojan Horse
Systweak	malware.keylogger
TACHYON	Trojan/W32.Keylogger.7014400
TrendMicro	TROJ_FR.7170E263
TrendMicro House Call	TROJ_FR.7170E263
VirusBlokAda	TrojanSpy.Agent
Zillya!	Trojan.Agent.Win32.1168788

YARA Rules

No matches found.

ssdeep Matches

No matches found.

PE Metadata

Compile Date	2017-03-26 00:28:24-04:00
Import Hash	0937a296014c778f116e3990f06e314b

PE Sections

MD5	Name	Raw Size	Entropy
a9fb26d3d4f4a80f2c2f7aeb1201325a	header	1024	3.391911
c788578d4f02ac011ffabd20db4506f3	.text	1619456	6.522579
7a1b03c4f7501d6f82d34a01fe9cf6b7	.rdata	348160	5.245418
50c4f4eab880975227b9b4d454941979	.data	24064	4.732755
b9af73df5ec7fb7a68b1c00d83e6b404	.gfids	111104	4.230152
52f93ebec3bc0c9da8e85ddf5ad812f4	.giats	512	0.155178
1f354d76203061bfd5a53dae48d5435	.tls	512	0.020393
e0376d74c0a0f746949b4647d35ef424	.rsrc	4774400	5.470347
9011be24e5ab8066360bd7d0af07cea6	.reloc	135168	6.491093

Packers/Compilers/Cryptors

Description

This sample is a GUI implant controller titled "Cyber Agent v11.0". It is capable of dynamically building new bot payloads with the following options:

```

-----Begin Payload Options-----
Callback IP
Callback Port
Beacon Interval
Output Path
-----End Payload Options-----
    
```

victim_info (see analysis for "43193c4efa8689ff6de3fb18e30607bb941b43abb21e8cee0cfd664c6f4ad97c") is displayed for each implant beacon received. The controller can establish Remote Desktop viewer, drive enumeration, file upload/download, list processes and services, reverse shell, microphone capture and recording, keylogger, browser activity, cached passwords, and DLL loading and unloading. The controller has the ability to provide implants with an Update URL as well as an option to uninstall all bots.

159.100.250.231

Ports

- 80 TCP
- 8080 TCP

Whois

% IANA WHOIS server

% for more information on IANA, visit <http://www.iana.org>

% This query returned 1 object

refer: whois.arin.net

inetnum: 159.0.0.0 - 159.255.255.255

organisation: Administered by ARIN

status: LEGACY

whois: whois.arin.net

changed: 1993-05

source: IANA

whois.arin.net

NetRange: 159.100.0.0 - 159.101.255.255

CIDR: 159.100.0.0/15

NetName: RIPE-ERX-159-100-0-0

NetHandle: NET-159-100-0-0-1

Parent: NET159 (NET-159-0-0-0-0)

NetType: Early Registrations, Transferred to RIPE NCC

OriginAS:

Organization: RIPE Network Coordination Centre (RIPE)

RegDate: 2003-10-29

Updated: 2003-10-29

Comment: These addresses have been further assigned to users in

Comment: the RIPE NCC region. Contact information can be found in

Comment: the RIPE database at <http://www.ripe.net/whois>

Ref: <https://rdap.arin.net/registry/ip/159.100.0.0>

ResourceLink: <https://apps.db.ripe.net/search/query.html>

ResourceLink: whois.ripe.net

OrgName: RIPE Network Coordination Centre

OrgId: RIPE

Address: P.O. Box 10096

City: Amsterdam

StateProv:

PostalCode: 1001EB

Country: NL

RegDate:

Updated: 2013-07-29

Ref: <https://rdap.arin.net/registry/entity/RIPE>

ReferralServer: [whois://whois.ripe.net](http://whois.ripe.net)

ResourceLink: <https://apps.db.ripe.net/search/query.html>

OrgTechHandle: RNO29-ARIN
OrgTechName: RIPE NCC Operations
OrgTechPhone: +31 20 535 4444
OrgTechEmail: hostmaster@ripe.net
OrgTechRef: <https://rdap.arin.net/registry/entity/RNO29-ARIN>

OrgAbuseHandle: ABUSE3850-ARIN
OrgAbuseName: Abuse Contact
OrgAbusePhone: +31205354444
OrgAbuseEmail: abuse@ripe.net
OrgAbuseRef: <https://rdap.arin.net/registry/entity/ABUSE3850-ARIN>

whois.ripe.net

inetnum: 159.100.245.0 - 159.100.255.255
netname: Akenes
descr: Exoscale Open Cloud DK2
descr: Exoscale cloud hosting <https://www.exoscale.ch>
descr: *****
descr: * These IPs are customer assigned STATIC IPs.
descr: * In case of abuse, please do NOT block entire
descr: * network as IPs of this block are assigned as /32
descr: * to individual customers.
descr: *****
descr: * For abuse-complaints please use
descr: * only abuse@exoscale.ch.
descr: *****
country: CH
admin-c: AC22866-RIPE
tech-c: LLL1007-RIPE
status: LEGACY
mnt-by: Exoscale-MNT
created: 2017-11-20T10:37:49Z
last-modified: 2017-11-20T10:37:49Z
source: RIPE

person: Antoine COETSIER
address: Boulevard de Grancy 19A
address: 1006 Lausanne
address: SWITZERLAND
phone: +41 58 255 00 66
nic-hdl: AC22866-RIPE
mnt-by: Exoscale-MNT
created: 2013-02-08T14:10:06Z
last-modified: 2019-04-11T05:30:08Z
source: RIPE # Filtered

person: Loic Lambiel
address: Boulevard de Grancy 19A
address: 1006 Lausanne
address: Switzerland
phone: +41 58 255 00 66
nic-hdl: LLL1007-RIPE
mnt-by: Exoscale-MNT
created: 2013-02-15T10:16:52Z
last-modified: 2019-04-11T05:31:04Z
source: RIPE # Filtered

% Information related to '159.100.248.0/21AS61098'

route: 159.100.248.0/21
origin: AS61098
mnt-by: Exoscale-MNT

created: 2016-12-14T10:12:52Z
 last-modified: 2016-12-14T10:12:52Z
 source: RIPE

% This query was served by the RIPE Database Query Service version 1.95.1 (WAGYU)

Relationships

159.100.250.231	Connected_From	1ea6b3e99bbb67719c56ad07f5a12501855068a4a866f92db8dcdefaffa48a39
159.100.250.231	Connected_From	b6811b42023524e691b517d19d0321f890f91f35ebddf1c12cbb92cda5b6de32
159.100.250.231	Connected_From	738ba44188a93de6b5ca7e0bf0a77f66f677a0dda2b2e9ef4b91b1c8257da790
159.100.250.231	Connected_From	43193c4efa8689ff6de3fb18e30607bb941b43abb21e8cee0cfd664c6f4ad97c

Description

Hard-coded C2 address used by these RATs.

43193c4efa8689ff6de3fb18e30607bb941b43abb21e8cee0cfd664c6f4ad97c

Tags

keyloggerspywaretrojan

Details

Name	83833f8dbdd6ecf3a1212f5d1fc3d9dd
Size	905216 bytes
Type	PE32 executable (GUI) Intel 80386, for MS Windows
MD5	83833f8dbdd6ecf3a1212f5d1fc3d9dd
SHA1	77a2272633eb64e4c16f8ea4466dba59ecc92292
SHA256	43193c4efa8689ff6de3fb18e30607bb941b43abb21e8cee0cfd664c6f4ad97c
SHA512	cda12a75b1d6524fe8856d6ef359ab58785e2c56ca4fec613b851a6730d24b8141dfdd00fba62f2865b8cc4606e85b258c02d71ccd45fcde7
ssdeep	24576:AECw5N98knVurfj9gbYX91XdKo1ldrtD9:AECwz9fqfj59NwuldrF
Entropy	6.710436

Antivirus

Ahnlab	Trojan/Win32.KeyLogger
Antiy	Trojan/Win32.AGeneric
Avira	HEUR/AGEN.1038092
BitDefender	Gen:Variant.Graftor.679285
ClamAV	Win.Trojan.Agent-7376468-0
ESET	a variant of Win32/Spy.Agent.PUH trojan
Emsisoft	Gen:Variant.Graftor.679285 (B)
Ikarus	Trojan-Spy.Agent
K7	Spyware (00555d821)
NANOAV	Trojan.Win32.Graftor.ggzicq
Sophos	Troj/Agent-BCXS

Symantec	Heur.AdvML.B
VirusBlokAda	BScope.TrojanSpy.Agent
Zillya!	Trojan.Agent.Win32.1170395

YARA Rules

No matches found.

ssdeep Matches

No matches found.

PE Metadata

Compile Date	2008-01-17 10:34:19-05:00
Import Hash	3b7df90688bca84764a888c49f25e8b9

PE Sections

MD5	Name	Raw Size	Entropy
064a795c4019629fd03c3d47c823cd49	header	1024	3.330520
ec60b9f4b78b0f79ea9d15910baf3d8d	.text	672768	6.660080
3dd902a53e33d4f6b014f6a677620252	.rdata	164864	5.832569
0c88a9a99d1c3cb1b61009a6acb2539e	.data	37376	5.304517
1f354d76203061bfd5a53dae48d5435	.tls	512	0.020393
d5ea2a2452a9733e2cc63487e98b387d	.gfids	512	2.821174
f42c4819230ff4b40b0e52850c134b08	.rsrc	512	4.708237
a1862d52a23162d56421552f09f1ca85	.reloc	27648	6.587842

Packers/Compilers/Cryptors

Relationships

43193c4efa...	Contained_Within	1ea6b3e99bbb67719c56ad07f5a12501855068a4a866f92db8dcdefaffa48a39
43193c4efa...	Connected_To	159.100.250.231

Description

This sample is a full-featured RAT executable contained within "1EA6B3E99BBB67719C56AD07F5A12501855068A4A866F92DB8DCDEFACFA48A39".

See Figure 1 for full list of commands a hardcoded C2 address of 159.100.250.231 on port 8080 is contained within the sample. The RAT is controllable by CAgent.exe variant "618A67048D0A9217317C1D1790AD5F6B044EAA58A433BD46EC2FB9F9FF563DC6".

The Imports are obfuscated by prepending "CARAT_" to the API names.

Packets are formatted in the following format:

```
-----Begin Packet Formatting-----
[OPCODE] [4 Bytes length of data] [data]
-----Begin Packet Formatting-----
```

Packets are encoded by performing an XOR on the data after the header with the XOR key 0x07. The implant initiates callback to C2, then immediately sends its victim_info.

-----Begin Victim_Info-----

- Language
- Country
- Victim_ID
- Computer_Name
- User_Name
- Implant_Version = "11.0"
- Victim_IP
- System_Architecture
- Drive_Letters
- OS_Version

-----End Victim_Info-----

133820ebac6e005737d5bb97a5db549490a9f210f4e95098bc9b0a7748f52d1f

Tags

trojan

Details

Name	a21171923ec09b9569f2baad496c9e16
Size	922624 bytes
Type	PE32 executable (GUI) Intel 80386, for MS Windows
MD5	a21171923ec09b9569f2baad496c9e16
SHA1	35ba8e39e6c8234ad55baf27130bb696179b7681
SHA256	133820ebac6e005737d5bb97a5db549490a9f210f4e95098bc9b0a7748f52d1f
SHA512	c1775b68b6b083323780150f6da654c6bcacf313b298fd243047402a0d0ec5631f8c90ed7ccc28ff4c1eaf2666e671b9c0f6bc068ca9e06557
ssdeep	12288:KsukuhRC+VmUmEViUUwsaXpx3U09S5j4J6dxLqm1JaSjyQiEyDlZk7SxTmgaA6i:pukuhRC+Vr24v3qhdDaSuQCBZk7SU/
Entropy	6.678910

Antivirus

Ahnlab	Malware/Win32.Generic
Antiy	Trojan/Win32.AGeneric
Avira	HEUR/AGEN.1038092
ClamAV	Win.Trojan.Agent-7376468-0
ESET	a variant of Win32/Agent.SSC trojan
Symantec	Heur.AdvML.B

YARA Rules

No matches found.

ssdeep Matches

No matches found.

PE Metadata

Compile Date	2017-03-26 09:21:10-04:00
Import Hash	80e9b5b96cb30be08b9f46dcd40ca0b6

PE Sections

MD5	Name	Raw Size	Entropy
480ee7622ef011b56ad9be1f520b53bb	header	1024	3.124211
e0689d923085269b1433eb46c62b9aad	.text	698880	6.634137
e1d4d4f7c07cb01481a7f937c1a399c5	.rdata	154112	5.641674
5b25e16d6a60901096dd38e8d609656f	.data	38912	5.185811
1f354d76203061bfd5a53dae48d5435	.tls	512	0.020393
4d9e4bd9bce353817d7013e17254399	.rsrc	512	4.717679
6c01df76342b581365053b6550340347	.reloc	28672	6.610094

Packers/Compilers/Cryptors

Relationships

133820ebac...	Contained_Within	b6811b42023524e691b517d19d0321f890f91f35ebddf1c12cbb92cda5b6de32
---------------	------------------	--

Description

This sample is a full-featured RAT executable contained within "B6811B42023524E691B517D19D0321F890F91F35EBBDF1C12CBB92CDA5B6DE32".

See analysis for file "1ea6b3e99bb67719c56ad07f5a12501855068a4a866f92db8dcdefaffa48a39" for additional details. This sample varies slightly in the following ways.

```
-----Begin Packet Formatting-----
[OPCODE][4 Bytes data length][4 Bytes unused][AUTH CODE 72 50 BF 9E][Data]
-----End Packet Formatting-----
```

The implant initiates callback to C2, then waits for tasking (DOES NOT immediately send its victim_info) and the Victim_info for this version contains Unicode strings, it additionally adds UserGeoID to victim_info.

The sample attempts to connect to 159.100.250.231:8080 4 times, with 1 minute between attempts. If does not succeed, then attempts to connect to www.example.com 4 times, with 1 minute between attempts. This loop continues until a connection is made.

Relationship Summary

1ea6b3e99b...	Connected_To	159.100.250.231
1ea6b3e99b...	Contains	43193c4efa8689ff6de3fb18e30607bb941b43abb21e8cee0cfd664c6f4ad97c
b6811b4202...	Connected_To	159.100.250.231
b6811b4202...	Contains	133820ebac6e005737d5bb97a5db549490a9f210f4e95098bc9b0a7748f52d1f
738ba44188...	Connected_To	159.100.250.231
159.100.250.231	Connected_From	1ea6b3e99bb67719c56ad07f5a12501855068a4a866f92db8dcdefaffa48a39
159.100.250.231	Connected_From	b6811b42023524e691b517d19d0321f890f91f35ebddf1c12cbb92cda5b6de32
159.100.250.231	Connected_From	738ba44188a93de6b5ca7e0bf0a77f66f677a0dda2b2e9ef4b91b1c8257da790
159.100.250.231	Connected_From	43193c4efa8689ff6de3fb18e30607bb941b43abb21e8cee0cfd664c6f4ad97c
43193c4efa...	Contained_Within	1ea6b3e99bb67719c56ad07f5a12501855068a4a866f92db8dcdefaffa48a39
43193c4efa...	Connected_To	159.100.250.231
133820ebac...	Contained_Within	b6811b42023524e691b517d19d0321f890f91f35ebddf1c12cbb92cda5b6de32

Mitigation

Displayed below is a Python3 script used to decrypt and extract the embedded files:

--Begin Decryption and Extraction Python3 Script--

```
import argparse
import struct

def truncate_nullterm_str(data):
    null_index = data.find(b'\x00')
    truncated_str = data[:null_index].decode('utf-8')
    return truncated_str

def decode(offset,buffer,length,key1,key2):
    dec = b''
    k3 = key1
    key1 = key1 >> 1
    while length > 0:
        k1 = key1
        k2 = key2
        dec += bytes([(buffer[offset] ^ k1 ^ k2 ^ k3) & 0xff])
        key1 = (key1 >> 8 | ((key1 * 8 ^ key1) & 0x7f8) << 0x14) & 0xffffffff
        k3 = (k3 & k2 ^ (k2 ^ k3) & k1)
        key2 = (key2 >> 8 | (((key2 * 2 ^ key2) << 4 ^ key2) & 0xfffff80 ^ key2 << 7) << 0x11) & 0xffffffff
        offset += 1
        length -= 1
    return dec

offset = 0
def parse_options(buffer):
    options = list(struct.unpack('I'*30, buffer[0:120]))
    options.append(buffer[120:320])
    options.append(buffer[320:2820])
    options.append(buffer[2820:3020])
    options.append(buffer[3020:3120])
    options.append(buffer[3120:3220])
    options.append(buffer[3220:3320])
    options.append(buffer[3320:3360])

    enabled_options = ''
    disabled_options = ''

    if options[0] == 0:
        global offset
        offset = options[27] + options[28] + options[22]
        enabled_options += "Embedded payload at offset: %d\n" % offset
        disabled_options += "Download payload\n"
    else:
        enabled_options += "Download payload from: %s\n" % truncate_nullterm_str(options[30])
        disabled_options += "Embedded payload\n"

    str = "VM Detect\n"
    if options[1] == 0:
        disabled_options += str
    else:
        enabled_options += str

    str = "Sandbox Detect\n"
    if options[2] == 0:
        disabled_options += str
    else:
        enabled_options += str

    str = "Debugger Detect\n"
    if options[3] == 0:
        disabled_options += str
```

```
else:
    enabled_options += str

str = "Active Processes Check\n"
if options[4] == 0:
    disabled_options += str
else:
    enabled_options += str

str = "Installed programs Check\n"
if options[5] == 0:
    disabled_options += str
else:
    enabled_options += str

str = "Language Check\n"
if options[6] == 0:
    disabled_options += str
else:
    enabled_options += str

str = "Username Check\n"
if options[7] == 0:
    disabled_options += str
else:
    enabled_options += str

str = "Computer name Check\n"
if options[8] == 0:
    disabled_options += str
else:
    enabled_options += str

str = "Installed number of programs Check\n"
if options[9] == 0:
    disabled_options += str
else:
    enabled_options += str

str = "Number running processes Check\n"
if options[10] == 0:
    disabled_options += str
else:
    enabled_options += "Number running processes Check: %d\n" % options[10]

str = "System processors/memory/diskspace Check\n"
if options[11] == 0:
    disabled_options += str
else:
    if options[11] & 0x001:
        enabled_options += "Processor count check: %d\n" % options[12]
    if options[11] & 0x010:
        enabled_options += "Physical memory check: %d\n" % options[13]
    if options[11] & 0x000:
        enabled_options += "Disk space check: %d\n" % options[14]

str = "DLL Hijack cliconfg.exe\n"
if options[12] == 0:
    disabled_options += str
else:
    enabled_options += str

str = "EnableLUA\n"
if options[16] == 0:
```

```
    disabled_options += str
else:
    enabled_options += str

str = "Create Persistence\n"
if options[17] == 0:
    disabled_options += str
elif options[17] == 1:
    enabled_options += "Create Persistence using Run key: %s\n" % truncate_nullterm_str(options[31][0x960:])
elif options[17] == 2:
    enabled_options += "Create Persistence in Startup folder\n"
elif options[17] == 3:
    enabled_options += "Create Persistence using \"System Backup\" hourly Scheduled Task\n"

if options[18] == 0:
    disabled_options += "Direct Execution\n"
else:
    disabled_options += "Process Hollowing\n"

if options[19] == 0:
    enabled_options += "Process Hollowing: self\n"
elif options[19] == 1:
    enabled_options += "Process Hollowing: svchost.exe\n"
elif options[19] == 2:
    enabled_options += "Process Hollowing: conhost.exe\n"
elif options[19] == 3:
    enabled_options += "Process Hollowing: explorer.exe\n"
elif options[19] == 4:
    enabled_options += "Process Hollowing: %httpshell\open\command\ registry key value\n"
elif options[19] == 5:
    enabled_options += "Process Hollowing: %s\n" % truncate_nullterm_str(options[33])

str = "Sleep Timer\n"
if options[20] == 0:
    disabled_options += str
else:
    enabled_options += "Sleep Timer: %d\n" % options[20]

str = "Kill Timer\n"
if options[21] == 0:
    disabled_options += str
else:
    enabled_options += "Kill Timer: %d\n" % options[26]

if options[29] == 0:
    enabled_options += "Relocate to: C:\\"
elif options[29] == 1:
    enabled_options += "Relocate to: %windir%\\"
elif options[29] == 2:
    enabled_options += "Relocate to: %system%\\"
elif options[29] == 3:
    enabled_options += "Relocate to: %programfiles%\\"
elif options[29] == 4:
    enabled_options += "Relocate to: %programfiles%\Common Files\"
elif options[29] == 5:
    enabled_options += "Relocate to: C:\\ProgramData\"
elif options[29] == 6:
    enabled_options += "Relocate to: %userprofile%\\"
elif options[29] == 7:
    enabled_options += "Relocate to: %userprofile%\Documents\"
elif options[29] == 8:
    enabled_options += "Relocate to: %temp%\\"
```

```

elif options[29] == 9:
    enabled_options += "Relocate to: %userprofile%\Favorites\\"
elif options[29] == 10:
    enabled_options += "Relocate to: %appdata%\"
elif options[29] == 11:
    enabled_options += "Relocate to: %localappdata%\"
if len(truncate_nullterm_str(options[34])) > 0:
    enabled_options += "%s\" % truncate_nullterm_str(options[34])
enabled_options += "%s\n" % truncate_nullterm_str(options[35])

str = "Mutex\n"
if len(truncate_nullterm_str(options[36])) == 0:
    disabled_options += str
else:
    enabled_options += "Mutex: %s\n" % truncate_nullterm_str(options[36])

print("\nDisabled Options:")
print(disabled_options)

print("\nEnabled Options:")
print(enabled_options)

def main():
    parser = argparse.ArgumentParser()
    parser.add_argument('filename')
    args = parser.parse_args()

    with open(args.filename, 'rb') as f:
        exe = f.read()
        PE_header_pos = struct.unpack('<i', exe[0x3c:0x3c+4])[0]
        PE_header_len = struct.unpack('<i', exe[PE_header_pos+0x54:PE_header_pos+0x54+4])[0]
        PE_header_length = struct.unpack('<h', exe[PE_header_pos+0x14:PE_header_pos+0x14+2])[0]
        section_headers_pos = PE_header_pos + PE_header_length + 0x18
        num_headers = struct.unpack('<h', exe[PE_header_pos+0x6:PE_header_pos+0x6+2])[0]
        curr_header_pos = section_headers_pos
        bitmap_pos = PE_header_len
        for i in range(num_headers):
            header_len = struct.unpack('<i', exe[curr_header_pos+0x10:curr_header_pos+0x10+4])[0]
            bitmap_pos += header_len
            curr_header_pos += 0x28
        key1 = struct.unpack('<I', exe[bitmap_pos+0x3a:bitmap_pos+0x3a+4])[0]
        bitmap_len = len(exe) - bitmap_pos
        bitmap_header_len = struct.unpack('<H', exe[bitmap_pos+0x3e:bitmap_pos+0x3e+2])[0]
        key2 = struct.unpack('<I', exe[bitmap_pos+0x36:bitmap_pos+0x36+4])[0]
        bitmap_len -= bitmap_header_len
        bitmap_len -= 0x036
        print("[ ] Decoding %d Bytes with:" % bitmap_len)
        print("   Key1: %s" % hex(key1))
        print("   Key2: %s" % hex(key2))
        dec = decode(0,exe[bitmap_pos+bitmap_header_len+0x36:],bitmap_len,key1,key2)
        print("[+] Decoding Complete!")
        parse_options(dec[0:0xd56-0x36])
        payload_pos = 0xd56-0x36+offset
        print("[ ] Found embedded payload, extracting..")
        with open(args.filename + "_payload.exe", 'wb') as out:
            out.write(dec[payload_pos:])
        print("[+] Wrote %d Bytes to %s" % (len(dec[payload_pos:]), args.filename + "_payload.exe"))

if __name__ == '__main__':
    main()
--End Decryption and Extraction Python3 Script--

```

Recommendations

CISA recommends that users and administrators consider using the following best practices to strengthen the security posture of their organization's systems. Any configuration changes should be reviewed by system owners and administrators prior to implementation to avoid unwanted impacts.

- Maintain up-to-date antivirus signatures and engines.
- Keep operating system patches up-to-date.
- Disable File and Printer sharing services. If these services are required, use strong passwords or Active Directory authentication.
- Restrict users' ability (permissions) to install and run unwanted software applications. Do not add users to the local administrators group unless required.
- Enforce a strong password policy and implement regular password changes.
- Exercise caution when opening e-mail attachments even if the attachment is expected and the sender appears to be known.
- Enable a personal firewall on agency workstations, configured to deny unsolicited connection requests.
- Disable unnecessary services on agency workstations and servers.
- Scan for and remove suspicious e-mail attachments; ensure the scanned attachment is its "true file type" (i.e., the extension matches the file header).
- Monitor users' web browsing habits; restrict access to sites with unfavorable content.
- Exercise caution when using removable media (e.g., USB thumb drives, external drives, CDs, etc.).
- Scan all software downloaded from the Internet prior to executing.
- Maintain situational awareness of the latest threats and implement appropriate Access Control Lists (ACLs).

Additional information on malware incident prevention and handling can be found in National Institute of Standards and Technology (NIST) Special Publication 800-83, "**Guide to Malware Incident Prevention & Handling for Desktops and Laptops**".

Contact Information

Document FAQ

What is a MIFR? A Malware Initial Findings Report (MIFR) is intended to provide organizations with malware analysis in a timely manner. In most instances this report will provide initial indicators for computer and network defense. To request additional analysis, please contact CISA and provide information regarding the level of desired analysis.

What is a MAR? A Malware Analysis Report (MAR) is intended to provide organizations with more detailed malware analysis acquired via manual reverse engineering. To request additional analysis, please contact CISA and provide information regarding the level of desired analysis.

Can I edit this document? This document is not to be edited in any way by recipients. All comments or questions related to this document should be directed to the CISA at 1-844-Say-CISA or contact@mail.cisa.dhs.gov.

Can I submit malware to CISA? Malware samples can be submitted via three methods:

- Web: <https://malware.us-cert.gov>
- E-Mail: submit@malware.us-cert.gov
- FTP: <ftp://malware.us-cert.gov> (anonymous)

CISA encourages you to report any suspicious activity, including cybersecurity incidents, possible malicious code, software vulnerabilities, and phishing-related scams. Reporting forms can be found on CISA's homepage at www.us-cert.gov.