GENERAL DYNAMICS
Fidelis Cybersecurity Solutions

www.fidelissecurity.com
www.threatgeek.com
@FidSecSys
+1800.652.4020

Follow-Up #1 Fidelis Threat Advisory #1011

Intruder File Report- Sneakernet Trojan

January 31, 2014

Document Status: FINAL
Last Revised: 2014-01-31

Executive Summary

Previous General Dynamics Fidelis Cybersecurity Services (Fidelis) reporting, ref: Fidelis Threat Advisory (FTA) #1011 dated 15 Jan 2014, introduced a malware system comprised of multiple files that provided a means for intruders to discover and retrieve data from disparate computer systems via removable storage devices. The malware system consists of at least two Portable Executable (PE) files, one acting as a headquarters component and one acting as field unit or agent component. The headquarters component infects drives connected to its host system with the field unit component and retrieves data from the field unit on the infected drive's return to the headquarters host system. The field unit conducts reconnaissance and data collection in accordance with particular commands.

Continuing analysis solidified the headquarters component's Command and Control (C2) scheme. The malware receives commands from a locally stored encrypted file.

This report describes select malware functionality with some granularity, provides extended detail regarding the headquarters component's C2 functionality, provides additional means of defensive detection of this malware and describes some interesting aspects of the malware as a whole.

The Fidelis team updated Fidelis XPS™ advanced threat defense system with additional rules to reflect current analysis findings associated with this malware.

Forensic Analysis Findings

Basic Functionality

Previous reporting, ref: Fidelis Threat Advisory (FTA) #1011 dated 15 Jan 2014, introduced a malware system comprised of multiple files that reflected a means for intruders to discover and retrieve data from disparate computer systems via removable storage devices. Analysis of the system relied on the availability of two files named netsat.exe and netui3.dll. Netsat.exe functioned as a master application affording intruders the ability, in a selective and controlled manner, to infest removable devices with an agent application in the form of netui3.dll, aka

Users are granted permission to copy and/or distribute this document in its original electronic form and print copies for personal use. This document cannot be modified or converted to any other electronic or machine-readable form in whole or in part without prior written approval of General Dynamics Fidelis Cybersecurity Solutions Inc.

While we have done our best to ensure that the material found in this document is accurate, General Dynamics Fidelis Cybersecurity Solutions makes no guarantee that the information contained herein is error free.

Copyright © 2014 General Dynamics Fidelis Cybersecurity Solutions

setup35.exe, aka update.exe. Previous reporting likened netsat.exe as a headquarters application and netui3.dll as a field unit with the following basic functionality:

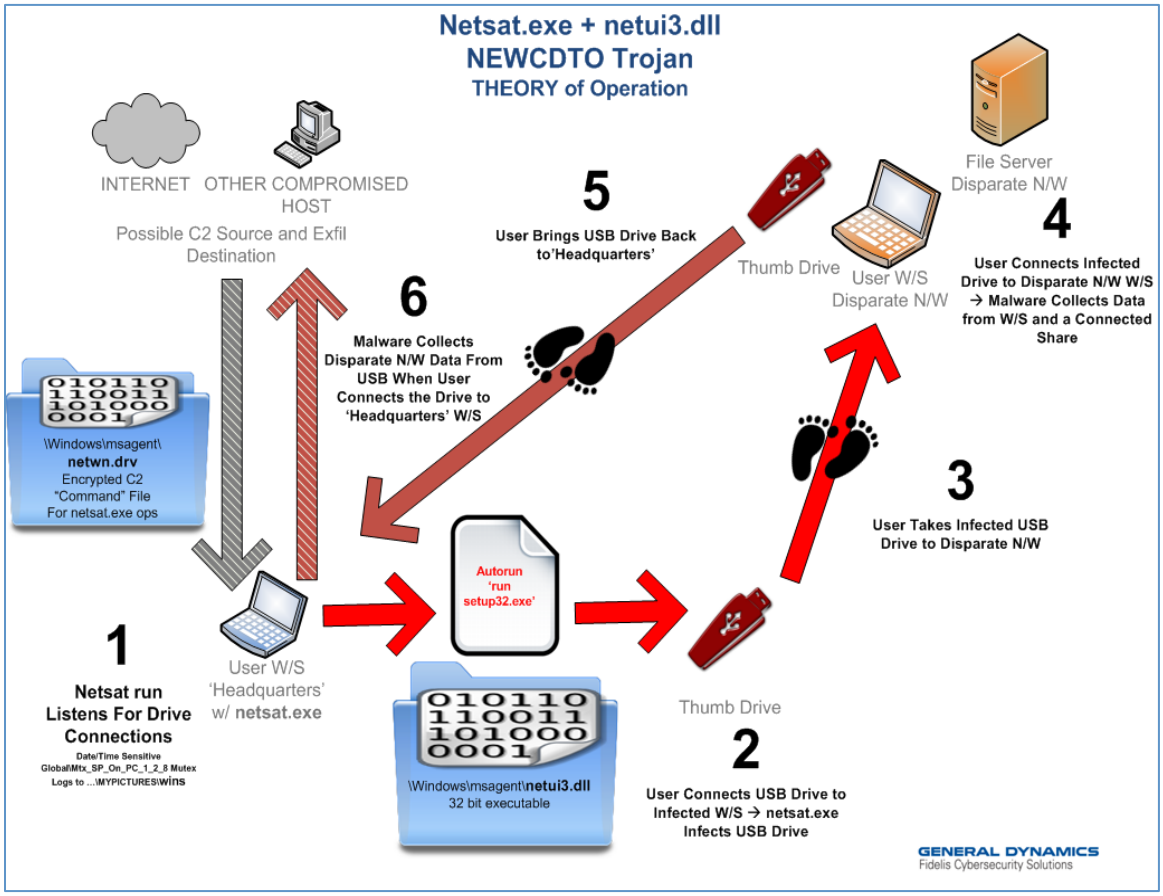
Headquarters (netsat.exe)

- Running on a possibly compromised system
- Logging some activity and errors to a file
- Receiving commands via an encrypted file on the local system (possible C2)
- Listening for drive connections
- Infecting connected drives with netui3.dll/winmgt.dll (setup35.exe + Autorun.inf)
- Collecting data gathered by any infected drives, ostensibly upon their return from being connected to other systems

Field Unit (netui3.dll)

- Collecting information about systems it comes into contact with through connection to the targeted systems with the drive whereon the malware resides
- Collecting file listings from local and share connected drives
- Discovering and connecting to shared drives visible to the local targeted system
- Copying and writing files to/from drives visible to the local targeted system

The following graphic serves to illustrate a possible basic theory of operation given available data:



Copyright © 2014 General Dynamics Fidelis Cybersecurity Solutions

Field Unit (netui3.dll/setup35.exe/update.exe) Functionality

File Name: netui3.dll

File Size: 39424 bytes

MD5: 68aed7b1f171b928913780d5b21f7617

Continued analysis disclosed details regarding the field unit/agent application. The following reflects observations during field unit execution from an infected external drive:

- The malware attempts to ensure errors are suppressed and not observed by a user
- The malware performs specific environment checking to adapt to Windows versions from at least Windows 2000 to Windows7/Server 2008 and up
- The malware terminates if it detects another iteration of itself via the Mutex "Mtx_Sp_on_PC_1_2_8"
- The malware terminates if any Gateway IPs associated with the resident system are in the 10.x.x.x range
- The malware copies itself to <CSIDL_LOCAL_APPDATA>\Microsoft\Windows\Help\update.exe
- The malware runs update.exe with the parameters "-wu *external drive letter*", e.g., z:, with the temporary directory specified for the working directory
- The malware copies a file named ~disk.ini from the infected drive to <CSIDL_LOCAL_APPDATA>\Microsoft\Windows\Help\intr
- The malware checks the system date against 31 May 2013; if on or after, the malware terminates
- The malware copies <CSIDL_LOCAL_APPDATA>\Microsoft\Windows\Help\intr to <CSIDL_LOCAL_APPDATA>\Microsoft\Windows\Chars\intr

Headquarters (netsat.exe) Functionality – C2 Mechanism

File Name: netsat.exe

File Size: 43520 bytes

MD5: eb8399483b55f416e48a320d68597d72

Previous analysis results indicated netsat.exe retrieved commands from an encrypted file named netwn.drv resident in the CSIDL_WINDOWS\msagent\ directory. The encryption was a Tiny Encryption Algorithm (TEA) implementation that used a key that was modified during encryption and decryption operations.

Copyright © 2014 General Dynamics Fidelis Cybersecurity Solutions

The following command file hex editor excerpt illustrates the command file's obfuscation in a contrived instance:

```
Offset     0  1  2  3  4  5  6  7   8  9 10 11 12 13 14 15
00000000   AA AA AA BE BA FE CA EF  BE AD DE 0D F0 AD 0B ED   aaa¾ºþÊï¾-Þ ð- í
00000016   FE DE C0 10 00 BB 6D E4  40 60 34 CC 6A 0A B7 2A   þÞÀ »mä@`4Ìj ·*
00000032   AA 43 C5 86 C6 10 00 FD  5B ED CE BE 6C D8 42 B4   ªCÅ†Æ ý[íÎ¾lØB´
00000048   90 AE 36 31 5D 40 A3 10  00 C0 5E 8A 4C 0F 0C 72    ®61]@£ À^ŠL  r
00000064   2E AA A2 28 20 16 20 0E  7A                       .ª¢(  z
```

Note: 1st Three Bytes = Unknown utility, ***Next 16 bytes*** *= Encryption Key, Bytes 20 and 21 = Command Data Size, Bytes 22-37 = encrypted command data*

Command Data Before Encryption/After Decryption

@@ d81596a9

ferry 0

dir 5

Analysis efforts did not have access to 'command' files retrieved from the victim systems for either the headquarters or the field unit applications. However, using the malware's behavior and determining the command file's format via reverse engineering afforded the ability to test numerous assumptions about the malware's intended use. Analysis determined the command format was: drive identification followed by one or more command and parameter strings. The following table reflects testing and theoretical contents of command files driving netsat.exe operation:

Copyright © 2014 General Dynamics Fidelis Cybersecurity Solutions

Test Commands One - Infection Attempt and Retrieval of Data Collection From a Remote System		
Command	**Description**	**Outcome**
@@ notmyser	designates a volume serial number	
ferry 1	infect the just listed identified drive	fails because notmyser does not match an attached drive
@@ d81596a9	designates a volume serial number	this S/N is from actually attached drive
getres	collect data harvested from a targeted system	success copies data from ext.drv\RECYCLED\RECYCLED\SYS to <CSIDL_NETHOOD>\Microsoft\Intel (Note: before copy checks if file exists in \Intel and determines file size - the implication is the possibility of updating previously retrieved files)
@@ 00	designates any connected drive	00 acts a wildcard for volume serial number
dir 5	retrieve a directory listing	did not execute because a volume serial number (d81596a9) was previously found
Test Commands Two - Retrieve Directory Listing From Any Connected Drive and Attempt Data Collection Retrieval		
Command	**Description**	**Outcome**
@@ 00	designates any connected drive	
dir 5	retrieve a directory listing	directory listing obtained from next connected drive
@@ d81596a9	designates a volume serial number	this S/N is from actually attached drive
getres	collect data harvested from a targeted system	did not execute

Copyright © 2014 General Dynamics Fidelis Cybersecurity Solutions

The Following Are Hypothetical Scenarios Designed to Illustrate Possible Employment Options	
Possible Commands One - Targeting Specific Devices (Known to Intruder From Previous netsat/netui3 Activity)	
Command	**Description**
@@ sernum1	designates a volume serial number
getres	collect data harvested from a targeted system
@@ sernum2	designates a volume serial number
ferry 1	infect this particular drive
@@ 00	designates any connected volume not listed above
dir 4	retrieve a directory listing from the just connected drive, re: 00
Possible Commands Two - Maximizing Propagation (Theoretical)	
Command	**Description**
@@ sernum1	designates a particular volume serial number
cmd1	particular command
cmd2	particular command
@@ sernum2	designates a particular volume serial number
cmd3	particular command
cmd4	particular command
cmd5	particular command
@@ 00	designates any connected volume not listed above
ferry 0	infect the just connected drive, re: 00
dir 5	retrieve a directory listing from the just connected drive, re: 00

Headquarters (netsat.exe) Functionality – Log File

The headquarters component (netsat.exe) logs certain events in a file located at CSIDL_MYPICTURES\wins. Analysis indicates the log file is probably stored in the clear, i.e., the contents are not obfuscated. Example log file contents are presented as follows:

St 01/18/13 12:03:30

into

d81596a9 ar 01/18/13 12:03:44

Total:30532M, Free:30387M

Copyright © 2014 General Dynamics Fidelis Cybersecurity Solutions

End copy : E:\RECYCLED\RECYCLED\SYS\file1.txt

End copy : E:\RECYCLED\RECYCLED\SYS\interesting.txt

Re on Fin

The following strings, which are not all inclusive or exclusive, could be used to find log files, fragments or contents on devices and on a network:

Format String	Example/Explanation
"Total:%I64dM, Free:%I64dM"	Total:30532M, Free:30387M
"!Get disk space"	
"error = %d"	error = 3
"!add drive, n = "	"!add drive, n = 5" (5 represents E drive)
"!u ser"	
"%08x ar %s %s"	<8 hex digits> ar <date> <time>
"Can't open file %s, error = %d"	
"ERROR Register notification"	
"!Up"	
"!ad dri, nD=%d"	"!ad dri, nD=5"
"!Cr Des\n"	Indicates failure to open desktop.ini for writing
"!Cr De.i. err=%d\n"	Specifies error code for failure to open desktop.ini for writing
"!up %s \n e:%d\n"	Indicates failure to create the RECYLCED/RECYCLER directory. Example: "!up E: <newline> e:3"
"Get Dir_c1 error!"	Indicates failure to retrieve that CSIDL_WINDOWS path for building of the netwi.drv path during ferry command

Copyright © 2014 General Dynamics Fidelis Cybersecurity Solutions

Format String	Example/Explanation
"!cp cf e:%d"	Indicates error copying netwi.drv to ~disk.ini.
"!c r\n"	Indicates error copying setup35.exe
"!c tr\n"	
"!c .inf, e:%d""	Indicates error opening AutoRun.inf for binary write
"!C c\n"	
"!c .inf"	Indicates could not create AutoRun.inf
"!c ser"	Indicates could not copy netu3.dll/setup35.exe
"!Get volume path = %s"	
"!Get disk memory"	

Interesting Artifacts and Observations

Previous and continuing analysis results indicated some interesting and/or relevant aspects of this malware:

- The malware tries to be quiet - error handling
- There was robust implementation intention across Windows versions
- The malware employs robust environment checking; frustrating inadvertent execution and analysis
- The malware prevents multiple iterations of itself on individual systems
- The malware does not run on systems using Gateways assigned a particular internal net range (10.x.x.x)
- The malware's execution has an expiration date
- The malware purposely obfuscates and complicates C2
- The malware injects complexity into C2 encryption operations; obfuscating execution and frustrating/delaying analysis
- The malware uses obscure file system paths
- The malware author, ironically or purposely, named a collected data storage folder 'Intel'

Copyright © 2014 General Dynamics Fidelis Cybersecurity Solutions

The following interesting questions/assumptions emerged from previous cursory analysis of this malware:

- C2 appears to be accomplished via providing commands in an encrypted file stored on the local 'master' system (re: netsat.exe). This C2 scheme would seem to dictate:
  - Intruder remote access to the 'master' system
  - Intruder local access to the 'master' system
  - a C2 delivery/retrieval component, such as another piece of code that downloads a C2 file
- Available information precludes determination of the means of exfiltration. Netsat.exe's data collection functionality suggested data destined for exfiltration might be collected by the 'master' system. This possibility suggests:
  - Intruder remote access to the 'master' system
  - Intruder local access to the 'master' system
  - An exfiltration mechanism in the form of another piece of code

Further analysis confirms the malware's use of an encrypted file stored on the system whereon the malware is executing without an apparent means of automatic generation. This continues to suggest that intruders either have local or remote access to headquarters systems running netsat.exe or access to another application that automates remote C2 data/file retrieval. Intruders' apparent ability to distinguish between particular field unit vehicles (infected drives), ref: Possible Commands One - Targeting Specific Devices (Known to Intruder From Previous netsat/netui3 Activity) from Hypothetical command table, suggests active engagement with the malware and targets.

Conclusion

This report is based on information extracted from reverse engineering and analysis of two PE files. There are other components and artifacts of this malware that are currently inaccessible to Fidelis analysts. Therefore, analysts extrapolated some of the behavior presented here. While analysts are confident about behaviors described to date, there could certainly be additional behaviors and nuances heretofore unseen.

Analysis of this malware continues to suggest that a sophisticated effort was behind its creation and employment. Actors went to great lengths to make the malware efficient and effective while building in obfuscation and complexity. Interesting artifacts and observations continue to be discovered and made, such as the malware's apparent expiration, the interesting naming convention for a directory to hold collected data, and the actors' apparent intention to avoid certain networks or network addressing schemes.

Copyright © 2014 General Dynamics Fidelis Cybersecurity Solutions

Analysis continues and any relevant additional information will be reported as soon as practicable.

Appendix 1 Commands (for reference purposes)

The following commands and their descriptions, listed by executable file, illustrate the submitted malware's functionality:

netsat.exe	
Command	**Description**
cpd	copies directories and contents
cpr	copies files with size checking
der	deletes files and records activity in log
dir	obtains a directory listing
ferry	writes malicious files to a hidden RECYCLED or RECYCLER directory Files: setup35.exe (renamed netui3.dll), Autorun.inf, ~disk.ini (renamed netwi.drv), act.te
getres	iteratively copies files from RECYCLED/RECYCLER directory on target drive, deletes from source after copy - source is assumed to be drive used to collect data from one or more systems

netui3.dll (setup35.exe)	
Command	**Description**
cp	copies files from one location to another
cpu	copies files from one location to another setting copied files as hidden
cptur	creates a directory and copies file to that directory
ddr	silently deletes directory (performs an FO_DELETE shell file operation on a directory with the FOF_NOERRORUI, FOF_NOCONFIRMATION, and FOF_SILENT flags set)
del	deletes a file
delu	deletes a file after setting attributes to normal
gd	recursively writes and reads encoded data to/from a directory
gdir	prints directory listings to ~FF323D.tmp; data gets encoded; original ~FF323D.tmp file is deleted
gf	writes and reads encoded data to/from a file

Copyright © 2014 General Dynamics Fidelis Cybersecurity Solutions

netui3.dll (setup35.exe)	
Command	**Description**
gfover	determines if it has access to a file; may be a temp file creation/rename involved
gi	collects system related and possibly network related information such as, domains, system information
ndr	creates a directory
newend	closes a file that was opened for writing
newstar	sets normal attributes on a targeted file, deletes the file, opens the same file name as a binary file
wr	writes a string to a new file opened by the newstar command.
runb	try to run a targeted executable and then checks for the existence of that file every second for the next 15 minutes as long as it exists
rune	try to run a targeted executable one time
slf	generates a targeted file listing, e.g., dir, then copies the files in the list one by one
srf	copies files in a list one by one
srmf	uses NetUseAdd to connect to ipc$ share of a target host, creates a listing of files in the c$ - z$ shares of the target host, copies the files to a new location, deletes the share connection added using NetuseAdd
Note: rows highlighted in grey denote a best guess on functionality; more analysis needed	

Copyright © 2014 General Dynamics Fidelis Cybersecurity Solutions