GENERAL DYNAMICS
Fidelis Cybersecurity Solutions

www.fidelissecurity.com
www.threatgeek.com
@FidSecSys
+1 800.652.4020

Fidelis Threat Advisory #1007

RECOVERING FROM SHAMOON

November 1, 2012

Document Status: FINAL
Last Revised: 2012-11-01

Executive Summary

The Shamoon malware has received considerable coverage in the past couple of months because of its destructive nature. Despite assertions that it is the work of amateurs, it has had a major impact on companies believed to have been affected. The basic functions of the malware are to infect, entrench, propagate, and wipe.

However because of the way the malware operates and how it is programmed to wipe, it can find itself being its own enemy. It will wipe data found in the Documents and Settings folder and the System32 folder, and then use a signed driver for disk access to start wiping at the disk level. Because the operating system needs certain files in the System32 folder to run, it was found that infected hosts will always restart before the malware can wipe completely at the disk level.

Due to this it was possible to make a complete recovery of Shamoon-infected file systems to the state they were in before the wiping made the OS unbootable and unreadable. In fact the majority of files outside of the System32 and Document and Settings folder are recoverable as well; this provided the opportunity for a successful and fruitful analysis, investigation, and remediation effort.

Threat Overview

According to community write-ups, the Shamoon malware appears to have been deployed against a couple of entities on or about August 15, 2012. The malware had self-propagating qualities and was designed to overwrite data on disks attached to or accessible from targeted systems. The malware's functionality, briefly summarized below, was covered in some detail in community postings, such as Kaspersky's Securelist blog. Analysis details and testing of an available sample of Shamoon by General Dynamics Fidelis Cybersecurity Solutions researchers revealed that the malware's wipe operations did not overwrite entire disks, but rather overwrote enough to prevent access to the affected file systems, along with substantial amounts of file data. However, analysis indicated that some files were still intact after the malware's write operations and subsequent system reboot.

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, Fidelis makes no guarantee that the information contained herein is error free.

Fidelis researchers surmised there might be a means of recovering file data from targeted systems from a forensic and investigative analysis point of view. With this goal in mind, researchers tested several possible ways of restoring disk data critical to the access of the targeted disk's file system. What follows is a brief description of what the sample of the Shamoon malware does and a description and results of researchers' file system recovery efforts.

Shamoon Wiper Functionality Actions:

- Executes a copy of itself as a scheduled job
- Deletes the file created for the scheduled job
- Entrenches itself as a service
- Execution of the entrenched file results in a dropped driver
- The dropped driver is loaded and executed
- The dropped driver facilitates disk access
- The malware overwrites disk data to include the contents of \\Documents and Settings (user data) and \\Windows\system32 (system data) directories
- The malware eventually overwrites the disk's boot records (Master Boot Record (MBR) and Volume Boot Record (VBR)) (Note: Testing was accomplished on disks with one partition)
- The malware appears to target user data first, then system data
- The nature of the overwrites is such that the malware writes only a certain amount of data to targeted files, starting at the files' beginning (Offset 0x0) and then writing a certain amount of data to other file locations
- Fidelis researcher observations included the following:
    - At some point during the writing (wiping) process, the targeted system tries to read file data that has been overwritten, prompting an attempt to restore the involved file
    - The system asks the user for media containing system files when it cannot find the system files it is looking for
    - The targeted system eventually reboots, resulting an error on restart because of the overwritten boot records
    - The disks targeted in testing were not completely overwritten; there was still apparently viable file data on the targeted disks
    - The result of the malware's operation was the prevention of accessing the targeted file system

Note: *The Shamoon sample Fidelis researchers had available looked very similar to that detailed in community write-ups. However, as of the date of publication, researchers were still analyzing the available sample. Therefore, differences between the available sample and others available to the community may become apparent in the future.*

Analysis and Testing Overview

Fidelis initially approached the Shamoon analysis strictly from a perspective of determining what forensic artifacts could be recovered from a targeted system. The goal was at least a partial

Copyright © 2012 General Dynamics Fidelis Cybersecurity Solutions

reconstruction of the events precipitating the Shamoon attack, and possibly using those events found on the targeted systems to determine a start of the attack, and a possible source. Analysis revealed the possibility that some user data would be recovered as a side benefit to the forensic analysis process.

Three types of operating systems were used for testing purposes; all testing occurred on laptops. The laptops were wiped, had the operating system installed, and then had the Shamoon malware executed on the system. The three operating systems used for testing were Windows XP, Windows 2003, and Windows 7. The malware executed with no issues except on Windows 7. The User Access Control (UAC) on the Windows 7 systems had to be turned off before the malware would execute and perform the wiping action as has been observed on other machines. This has been noted by others in the community as well, specifically that Administrator access is needed for initially launching Shamoon.

Shamoon operation results in much of the data on the affected systems being overwritten with the fragmented image of a burning flag. As has been detailed above, the wipe function will overwrite data within the Documents and Settings folder followed by the System32 folder, and then it will start the physical disk access and start the wiping at the disk level. If the system restarts before the malware has completed wiping the disk then much of the data can still be recovered: each of our tests showed the system did restart before the disk was completely wiped. The amount wiped from the host will never be the same from system to system, mainly because the size of the disk and partitions will all need to be taken into account.

VBR and File System Recovery Strategies

The following is the view of the wiped disk for each of the operating systems that we tested:

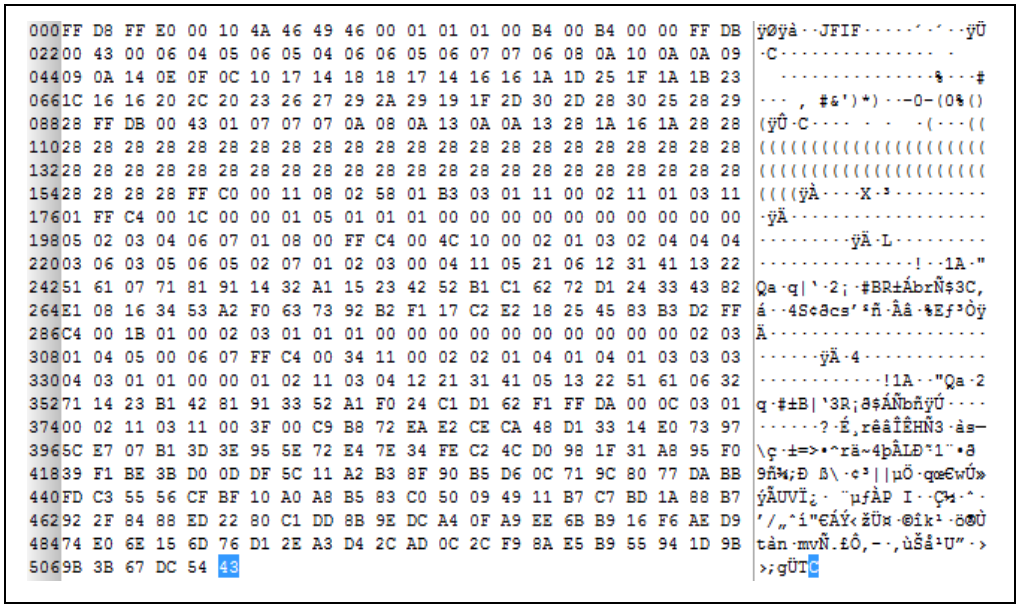


Fig 1. Example of wiped of MBR and VBR wiped by Shamoon Malware.

Figure 1 was found at the MBR (Sector 0) and the VBR (Sector 63/56 (XP, 2003), and 2048/206848 (7)) of each of the operating systems (As well as throughout the drive). Fidelis

Copyright © 2012 General Dynamics Fidelis Cybersecurity Solutions

researchers decided to look further into the drive and find if there was any possibility of recovering files or logs that would help illuminate what happened to the systems, and if any artifacts of the malware could be recovered.

Note on the VBR: *VBR stands for Volume Boot Record, and is made up of the boot sector and bootstrap code. The boot sector takes up 1 sector on the drive; the next 6 sectors on the drive are allocated for the bootstrap code. In all 16 sectors are allocated in total for the VBR. The VBR is created when a file system is created on a partition. In this paper we will be covering the NTFS Boot Record. The VBR is used to load machine code into RAM to start a program. Normally this program is the operating system.*

Keyword searches revealed that there were still files that would be recoverable on the system. In particular it was found that registry files and headers were still on the disk. After this, it was found that the Master File Table (MFT) was still, for the most part, intact. Trying to avoid the long and laborious process of carving files from the disk, researchers decided that it would instead be worth the time to try and recover the file system.

When the Windows operating system is installed or an NTFS volume created, a backup copy of the VBR is written to the last sector of the volume. This is a very important detail, as the forensic value of the VBR is substantial (See Figure 2). The area that will contain the critical information is known as the Bios Parameter Block (BPB). With this information it is possible to rebuild the file system as it existed before the wipe.

Copyright © 2012 General Dynamics Fidelis Cybersecurity Solutions

EB 52 90 4E 54 46 53 20 20 20 20 00 02 08 00 00 00 00 00 00 00 F8 00 00 3F 00 FF 00 3F 00 00 00 00 00 00 00 80 00 80 00 C0 F8 F8 0D 00 00 00 00 00 00 00 0C 00 00 00 00 00 8C 8F DF 00 00 00 00 00 F6 00 00 00 01 00 00 00 26 FA CA 70 02 CB 70 44 00 00 00 00

[… Truncated for size …]

00 00 55 AA

EB 52 90 – Instruction to jump to boot code (Not necessary for our application)

4E 54 46 53 20 20 20 20 – OEM Name (NTFS)

00 02 – Bytes per sector, 0x0200 = 512 Bytes.

08 – Sectors per cluster = 8

F8 – Media descriptor (Not necessary for our application)

C0 F8 F8 0D 00 00 00 00 – Total sectors in file system, 0x00DF8F8C0 = 234420416 Sectors (Add on the sector location of VBR for actual end of the file system, in this example the VBR is at sector 63 therefore the total sectors in the file system are 234420416 + 63 = 234420479)

00 00 0C 00 00 00 00 00 – Starting cluster of the MFT, 0X000C0000 = 786432 Clusters. 786432 * 8 (Cluster size) +63 (VBR Sector) = 6291519 Sectors

8C 8F DF 00 00 00 00 00 – Starting cluster of the MFT mirror, 0x00DF8F8C = 14651276. 14651276 * 8 + 63 = 117210271 Sectors

F6 – Size of MFT Entry, 246.

01 – Index size, 1.

26 FA CA 70 02 CB 70 44 – Serial number.

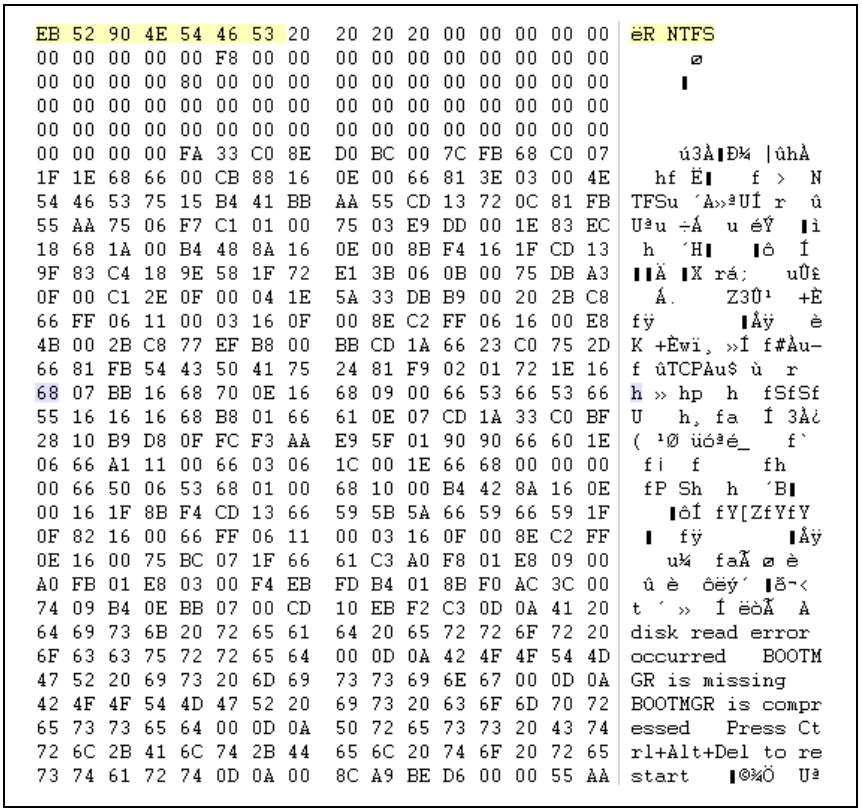
For more technical information on file systems and their forensic value, Brian Carrier's book File System Forensic Analysis is an invaluable tool.

Fig 2. Example of a broken down BPB found within the boot sector.

Just because the boot sector of the VBR is recoverable doesn't mean that everything on the file system will be restored to normal. If a file was wiped by the malware then it will still be wiped, or partially wiped. However files that weren't wiped will be much easier and faster to recover then carving and the context of each file will be easy to interpret.

To recover or identify the backup VBR a search will need to be run across the image file. It is preferable if the image file is a raw image as they are easier to edit then other image file formats. The search was performed for the hex of the VBR file header, EB 52 90 4E 54 46 53 (ëR□NTFS). A few hits were found throughout the drive, and it appeared that there were multiple empty VBR templates throughout the system (Shown in Figure 3). The correct VBR will likely be the one with information filled in from offset 10 – 80 (See Figure 2 to breakdown). During testing it was found that the last hit was normally the correct VBR, as this would be the VBR found at the end of the volume.

Copyright © 2012 General Dynamics Fidelis Cybersecurity Solutions



```
EB 52 90 4E 54 46 53 20  20 20 20 00 00 00 00 00  ëR NTFS
00 00 00 00 00 F8 00 00  00 00 00 00 00 00 00 00      ø
00 00 00 00 80 00 00 00  00 00 00 00 00 00 00 00     ▌
00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00
00 00 00 00 FA 33 C0 8E  D0 BC 00 7C FB 68 C0 07      ú3À▌Ð¼ |ûhÀ
1F 1E 68 66 00 CB 88 16  0E 00 66 81 3E 03 00 4E    hf Ë▌   f > N
54 46 53 75 15 B4 41 BB  AA 55 CD 13 72 0C 81 FB  TFSu ´A»ªUÍ r  û
55 AA 75 06 F7 C1 01 00  75 03 E9 DD 00 1E 83 EC  Uªu ÷Á  u éÝ  ▌ì
18 68 1A 00 B4 48 8A 16  0E 00 8B F4 16 1F CD 13   h  ´H▌   ▌ô  Í
9F 83 C4 18 9E 58 1F 72  E1 3B 06 0B 00 75 DB A3  ▌▌Ä ▌X rá;   uÛ£
0F 00 C1 2E 0F 00 04 1E  5A 33 DB B9 00 20 2B C8    Á.     Z3Û¹  +È
66 FF 06 11 00 03 16 0F  00 8E C2 FF 06 16 00 E8  fÿ        ▌Âÿ   è
4B 00 2B C8 77 EF B8 00  BB CD 1A 66 23 C0 75 2D  K +Èwï¸ »Í f#Àu-
66 81 FB 54 43 50 41 75  24 81 F9 02 01 72 1E 16  f ûTCPAu$ ù  r
68 07 BB 16 68 70 0E 16  68 09 00 66 53 66 53 66  h » hp  h  fSfSf
55 16 16 16 68 B8 01 66  61 0E 07 CD 1A 33 C0 BF  U   h¸ fa  Í 3À¿
28 10 B9 D8 0F FC F3 AA  E9 5F 01 90 90 66 60 1E  ( ¹Ø üó²é_   f`
06 66 A1 11 00 66 03 06  1C 00 1E 66 68 00 00 00   f¡  f     fh
00 66 50 06 53 68 01 00  68 10 00 B4 42 8A 16 0E   fP Sh  h  ´B▌
00 16 1F 8B F4 CD 13 66  59 5B 5A 66 59 66 59 1F    ▌ôÍ fY[ZfYfY
0F 82 16 00 66 FF 06 11  00 03 16 0F 00 8E C2 FF  ▌  fÿ       ▌Âÿ
0E 16 00 75 BC 07 1F 66  61 C3 A0 F8 01 E8 09 00    u¼  faÃ ø è
A0 FB 01 E8 03 00 F4 EB  FD B4 01 8B F0 AC 3C 00   û è  ôëý´ ▌ð¬<
74 09 B4 0E BB 07 00 CD  10 EB F2 C3 0D 0A 41 20  t ´ »  Í ëòÃ  A
64 69 73 6B 20 72 65 61  64 20 65 72 72 6F 72 20  disk read error
6F 63 63 75 72 72 65 64  00 0D 0A 42 4F 4F 54 4D  occurred   BOOTM
47 52 20 69 73 20 6D 69  73 73 69 6E 67 00 0D 0A  GR is missing
42 4F 4F 54 4D 47 52 20  69 73 20 63 6F 6D 70 72  BOOTMGR is compr
65 73 73 65 64 00 0D 0A  50 72 65 73 73 20 43 74  essed   Press Ct
72 6C 2B 41 6C 74 2B 44  65 6C 20 74 6F 20 72 65  rl+Alt+Del to re
73 74 61 72 74 0D 0A 00  8C A9 BE D6 00 00 55 AA  start   ▌©¾Ö  Uª
```

Fig 3. Example of a blank VBR.

Note: *On 2003 systems the boot backup is sometimes found halfway through the partition, a manual parsing of the file will need to be performed to confirm the VBR is legitimate for the partition.*

Once the VBR was found we noted the offset and calculated the sector to locate the backup in our desired forensic program. For the purposes of testing we used EnCase (v6.19.6). Once the sector of the backup VBR is known, EnCase was started and the image of the infected system was loaded. Within disk view we located the backup VBR and right clicked to add a manual partition as an NTFS file system. The partition was added, the MFT read, and the file system appeared:

Copyright © 2012 General Dynamics Fidelis Cybersecurity Solutions

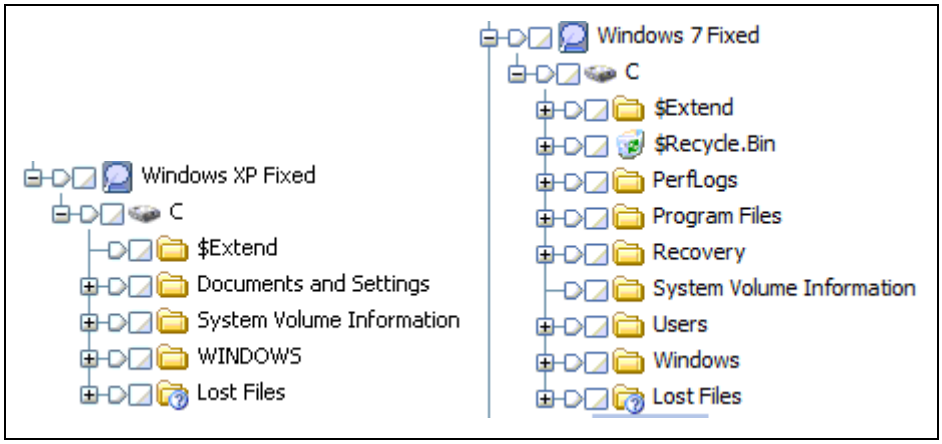


Fig 4. File system recovered within EnCase.

With the file system restored some relevant artifacts can be located now, and an actual computer forensic examination can take place.

This recovery can be successful without the use of the EnCase suite of forensic software as well. Using a hex editor of your choice to repair the image, in our testing WinHex (16.6) was used. Find the file header of the backup VBR within a file editor, copy from the header to footer of the boot sector; the footer will always be 55 AA. The size will be 512 bytes from header to footer. Then depending on what operating system is being examined you can write the copied boot sector to the appropriate sector on the affected image.

Placing the boot sector into the correct location will be the trickiest part as incorrect placement will result in the file system not being recognized. The boot sector should be placed at sector 63/56 for XP/2003, and at sector 2048/206848 for Windows 7. After this is complete you will be able to add the image into the forensic program. If the file system is not recognized then it is possible that the MBR will need to be reconstructed, though this is unlikely. Before rebuilding the MBR try adding the image as a volume and not as a disk.

Note: *Other recovery techniques are certainly viable as well. There are automated partition rebuilding tools available, though some of these rely on a valid MBR to work properly (In this case that wouldn't be feasible). Other options would be the fixboot command from the Windows Recovery Console found on a Windows OS disk. What we have presented here are forensically sound methods that are easily repeatable and least damaging to the evidence/image.*

Multiple Partition Recovery Strategies

For testing purposes the system with Windows 2003 was set up with three different partitions. We wanted to emulate the situation in which one would have multiple partitions on the computers, as is quite common. Conceivably the malware should wipe all of these partitions as well, as has been seen within the code of the malware. What we wanted to look at was the extent of the wiping on the partitions and whether the same techniques that were applied to a single partitioned drive would still apply on the multiple partitioned drive.

In theory each partition should be recoverable, as non-bootable partitions still create a VBR and place the backup at the end of the partition when a NTFS file system is installed. After searching

Copyright © 2012 General Dynamics Fidelis Cybersecurity Solutions

through the drive we found that there were three VBRs that all seemed to have corresponding information for the partitions that were originally created. On our test system we found that EnCase was not adding the partitions in a way that would recognize the file system as it did for the other systems. This could be because our boot sector was at sector 56 and not 63, or because the multiple partitions clash when trying to add them in.

We ended up having to edit the image by adding the backup boot sectors into the correct sector where the originals were found.

Partition	Boot Sector Placed At	Backup Boot Sector
1 (Primary)	56	41926079
2	41926080	62417879
3	62417880	82909679

Fig 5. VBR Placement in Windows 2003

Note: *The VBR placement for the next partition starts after the backup VBR of the preceding partition.*

Once the VBRs were added correctly we proceeded to add the partitions into EnCase.

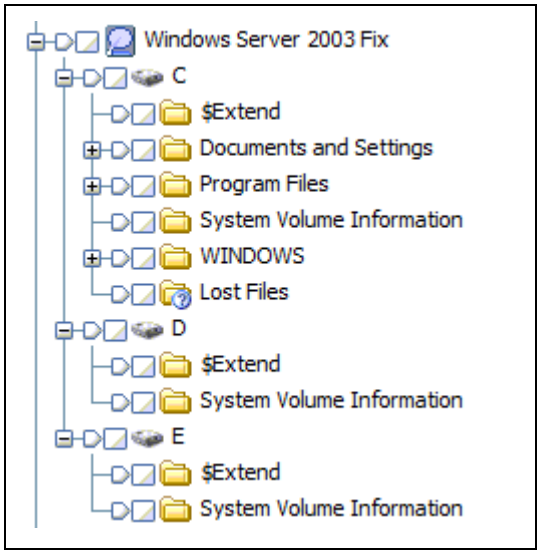


Fig 6. Reconstructed file system of a Windows 2003 operating system wiped by Shamoon.

Note: *EnCase gives default volume labels when added, so C, D, and E are respectively 1, 2, and 3.*

The extent of the wiping appeared to be on the same level to what was found on single partitioned drives. As mentioned before this was to be expected as the malware tries to wipe mounted and other volumes first and will then move to the primary volume (1/C).

Copyright © 2012 General Dynamics Fidelis Cybersecurity Solutions

The Fidelis Take

Fidelis researchers have developed a set of rules for detecting the Shamoon malware along the entire threat life cycle: initial infection, lateral propagation, and command and control communication. The embedded malware detection engine also recognizes the variant of Shamoon malware analyzed. All sensor configurations are capable of detecting the initial infection and the command and control communication, and the Fidelis XPS Internal Sensor is required for detecting the lateral movement of the malicious program.

Further Reading

- Shamoon the Wiper Copycats at Work (2012), retrieved 26 Oct 2012 from http://www.securelist.com/en/blog/208193786/Shamoon_the_Wiper_Copycats_at_Work
- Shamoon the Wiper in details, Tarakanov , Dmitry (2012), retrieved 26 Oct 2012 from http://www.securelist.com/en/blog/208193795/Shamoon_the_Wiper_in_details
- Shamoon the Wiper in details II, Tarakanov , Dmitry (2012), retrieved 26 Oct 2012 from http://www.securelist.com/en/blog/208193834/Shamoon_The_Wiper_further_details_Part_II
- Shamoon, a two-staged targeted attack (2012), retrieved 26 Oct 2012 from http://blog.seculert.com/2012/08/shamoon-two-stage-targeted-attack.html
- 'Shamoon' Virus Most Destructive Ever To Hit A Business, Leon Panetta Warns (2012), retrieved from http://www.huffingtonpost.com/2012/10/11/shamoon-virus-leon-panetta_n_1960113.html
- Carrier, Brian (2005). File System Forensic Analysis. Upper Saddle River, NJ: Pearson Education Inc.

Copyright © 2012 General Dynamics Fidelis Cybersecurity Solutions