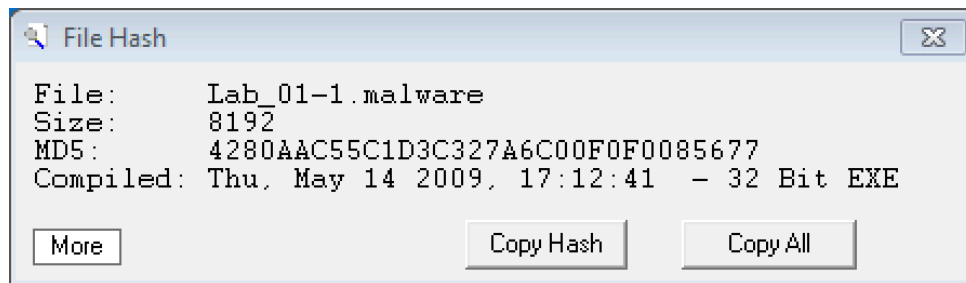


Lab 01-1.malware

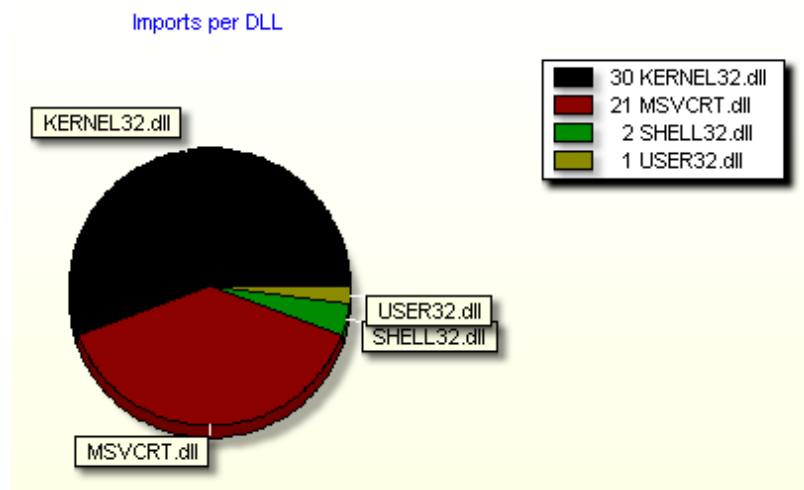
1. When was this file compiled?

This file was compiled on Thursday, May 14th, 2009.



2. List a few imports or sets of imports and describe how the malware might use them.

- I. *KERNEL32.dll* There seems to be a lot of multithreading code function stubs, process management functions and file I/O functions. This malware has the potential to impact the targeted system significantly.
- II. *MSVCRT.dll* This is the standard C library for the Visual Basic C++ Compiler versions 4.2 to 6.0. This could be used by malware which executes its payload using C or C++ code.
- III. *SHELL32.dll* This could be used by the malware for providing a remote shell.



A graph showing percentage of imports by DLL (produced by FileAlyzer)

3. What are a few strings that stick out to you and why?

- I. <http://www.ueopen.com/test.html> This URL was found in the `.data` section of the PE. The webpage is currently blank with a single javascript redirect function to a parked domain.
- II. `60.248.52.95:443` This IP address could be the domain of a command and control server used by the attacker. It is a Taiwanese IP address.
- III. `cmd.exe` This string appears in the `.data` section of the PE, indicating the malware executes shell commands (Note: It does launch a shell as a child process).

4. What happens when you run this malware? Is it what you expected and why?

When running this malware, nothing blatant appears on the screen. To an unsuspecting user, the only obvious action that happens is that the executable deletes itself.

Upon further inspection, the malware attempts to create a TCP session with the IP address in 3.III over HTTPS. Once it receives an ACK packet from the server, it presents a shell prompt to the attacker over the TCP connection, visible below in the packet data.

0000	00 50 56 34 37 a5 00 50 56 29 f9 14 08 00 45 00	.PV47..P V)....E.
0010	00 3e 15 0a 40 00 80 06 00 00 ac 10 02 06 3c f8	.>...@... ..<.
0020	34 5f c0 ff 01 bb 14 d8 67 d7 3f ab 82 3e 50 18	4_..... g.?..>P.
0030	01 00 1f 9e 00 00 2a 28 53 59 29 23 20 4d 41 4c*(SY)# MAL
0040	57 41 52 45 41 4e 41 4c 59 53 49 53	WAREANAL YSIS

The malware sends `*(SY)# MALWAREANALYSIS` to the attacker. `MALWAREANALYSIS` is the hostname of the infected machine.

Additionally, the malware launches a child process with a standard shell (`cmd.exe`) for the express purpose of deleting the original executable.

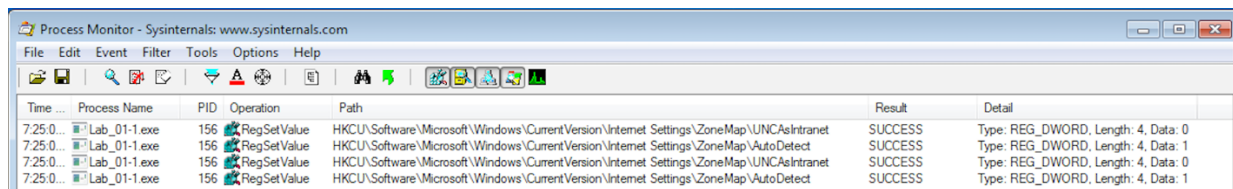
5. Name a procmon (Process Monitor) filter and why you used it?

Filtering of events by `Parent PID == $(PID of malware PE)`. This enabled me to see any child processes launched by the malware.

6. Are there any host based signatures? (Files, registry keys, processes or services, etc.). If so, what are they?

The malware changes two registry values (it does a blind-write to both, twice)

HKCU\Software\Microsoft\Windows\CurrentVersion\Internet Settings\ZoneMap\UNCAsIntranet: 0
 HKCU\Software\Microsoft\Windows\CurrentVersion\Internet Settings\ZoneMap\AutoDetect: 1



The screenshot shows the Process Monitor application window with the following data in the main pane:

Time ...	Process Name	PID	Operation	Path	Result	Detail
7:25:0...	Lab_01-1.exe	156	RegSet Value	HKCU\Software\Microsoft\Windows\CurrentVersion\Internet Settings\ZoneMap\UNCAsIntranet	SUCCESS	Type: REG_DWORD, Length: 4, Data: 0
7:25:0...	Lab_01-1.exe	156	RegSet Value	HKCU\Software\Microsoft\Windows\CurrentVersion\Internet Settings\ZoneMap\AutoDetect	SUCCESS	Type: REG_DWORD, Length: 4, Data: 1
7:25:0...	Lab_01-1.exe	156	RegSet Value	HKCU\Software\Microsoft\Windows\CurrentVersion\Internet Settings\ZoneMap\UNCAsIntranet	SUCCESS	Type: REG_DWORD, Length: 4, Data: 0
7:25:0...	Lab_01-1.exe	156	RegSet Value	HKCU\Software\Microsoft\Windows\CurrentVersion\Internet Settings\ZoneMap\AutoDetect	SUCCESS	Type: REG_DWORD, Length: 4, Data: 1

It also launches a shell process and runs the command

```
cmd.exe /c del \path\to\malware > null
```

which deletes the executable.

7. Are there any network based signatures? (URLs, packet contents, etc.) If so, what are they?

The malware attempts to initiate a TCP session over port 443 with the IP address 60.248.52.95. Upon receiving a response from the server, it sends a packet with a remote shell prompt `*(SY)# $HOSTNAME`.

8. Is there anything that impeded your analysis? How so? How might you overcome this?

The file deletes itself on every run. This can be overcome by creating a copy before executing, or by using a Snapshot-restore functionality on the Hypervisor.

9. What do you think is the purpose of the malware?

The purpose of this malware seems to be providing a backdoor/reverse-shell to an attacker on the infected machine.

Lab 01-2.malware

1. What is the md5sum? What of interest does VirusTotal report?

02658BC9801F98DFDF167ACCF57F6A36. VirusTotal reports this a trojan/downloader.

2. List a few imports or sets of imports and describe how the malware might use them.

- I. *KERNEL32.dll* Most of the imports from the kernel library relate to process creation or file I/O. This supports the claim from VirusTotal that this is a downloader.
- II. *WININET.dll* This library is used for higher-level internet protocols like HTTP and FTP. From the looks of the functions imported by this library, the malware primarily uses HTTP.
- III. *MSVCRT.dll* This is the standard C library for the Visual Basic C++ Compiler versions 4.2 to 6.0. This could be used by malware which executes its payload using C or C++ code.

3. What are a few strings that stick out to you and why?

- I. The file contains a whole section within the *.rsrc* part of the PE related to the “version info” of the executable. It seems to be filled in with generic Microsoft values. One thing that stands out is the name of the file appears to be *svchost.exe*, which could indicate that this malware attempts to hide itself by using a commonly running process name.
- II. *69.25.50.10* This could be the IP address of a C&C server run by the attacker.
- III. *Begin Downloader* This confirms the VirusTotal reports that claim this is a downloader/trojan.

4. What happens when you run this malware? Is it what you expected and why?

The malware sends a *CONNECT / HTTP/1.1* request to the IP address 69.25.50.10. This is expected, as the malware seems to be a downloader and *CONNECT* requests are used for creating tunneled, raw-data, TCP sessions through a proxy (the malware also changes many keys in the registry regarding proxies).

5. Name a procmon filter and why you used it.

Operation == RegSetValue. This filter allows us to see any changes that the malware makes to the system's registry.

6. Are there any host based signatures? (Files, registry keys, processes or services, etc.). If so, what are they?

Like *Lab_01-1.malware*, this piece of malware also touches the ZoneMap > UNCAsIntranet and Autodetect registry keys (setting them to 0 and 1, respectively). In addition, however, this malware also changes the Internet Settings > ProxyEnable registry key to be 0. The malware also launches the Windows Updater service as a child process.

7. Are there any network based signatures? (URLs, packet contents, etc.) If so, what are they?

The malware sends HTTPS CONNECT requests to 69.25.50.10.

8. Is there anything that impeded your analysis? How so? How might you overcome this?

Since the server at the IP address is no longer responding to requests, I had to use INetSim to fake an HTTPS server to see that the malware was attempting to create a tunnel through the HTTPS CONNECT request. Additionally, by controlling the server that was responding to the malware's requests, I was able to see the unencrypted data that the malware was sending.

9. What do you think is the purpose of the malware?

This malware appears to be a downloader. The strings in the *.data* section provide conclusive evidence of this, and can be seen below.

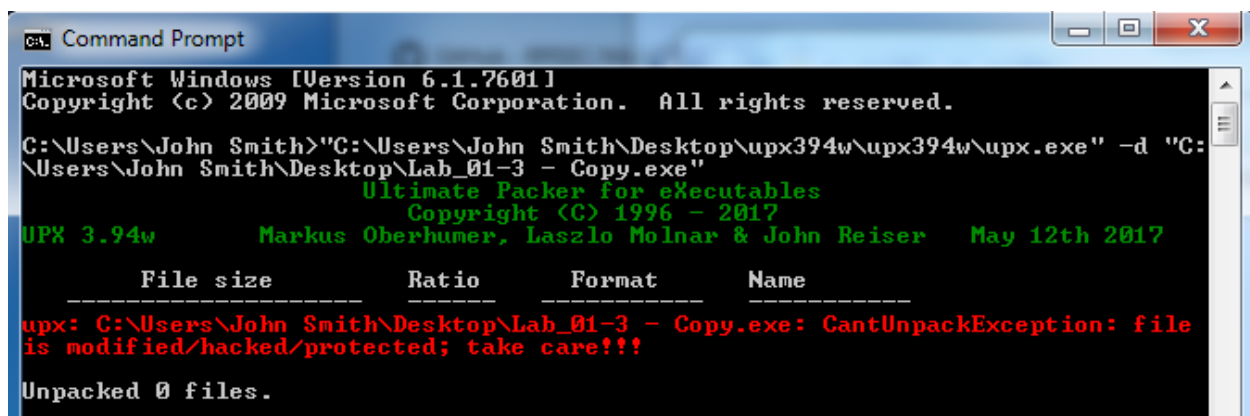
```
</head>.....<head>.....Begin Download..D-o-w-n-  
l-o-a-d-f-i-l-e%s*****%d@@@@@%d...Could not op  
en file for reading.....rb..wb..Begin Upload....  
U-p-l-o-a-d-f-i-l-e%s*****%d...eyb.exit....exit  
...Go on!..putf....use error! putf [transpeed]  
[filepath]..\...%s %d %s...getf....\tasks\.cmd  
/c .begin...Type command disable.Go on!.type....  
wuauc1t.exe.Hello.I am here!....CONNECT.HTTP/1.0
```

Lab 01-3.malware

1. Are there any indications that this malware is packed? What are they? What is it packed with?

- I. The PE section names are LOL0, LOL1, and .rsrc.
- II. The only imported functions are LoadLibrary, GetProcAddress, VirtualProtect/Alloc/Free, and ExitProcess. The first 3 are very common indicators of a packed program.
- III. PEiD reports this as being packed with UPX 0.89.6 – 1.02 / 1.05 – 2.90.

2. Are you able to unpack it? Why or why not?



```

Microsoft Windows [Version 6.1.7601]
Copyright (c) 2009 Microsoft Corporation. All rights reserved.

C:\Users\John Smith>"C:\Users\John Smith\Desktop\upx394w\upx394w\upx.exe" -d "C:\Users\John Smith\Desktop\Lab_01-3 - Copy.exe"
Ultimate Packer for eXecutables
Copyright (C) 1996 - 2017
UPX 3.94w Markus Oberhumer, Laszlo Molnar & John Reiser May 12th 2017

  File size      Ratio      Format      Name
-----
upx: C:\Users\John Smith\Desktop\Lab_01-3 - Copy.exe: CantUnpackException: file
is modified/hacked/protected; take care!!!

Unpacked 0 files.

```

UPX is unable to unpack it, stating the executable is “modified/hacked/protected.”

3. What are a few strings that stick out to you and why?

The following strings were pulled directly from memory using ProcessExplorer after launching the malware.

- I. *http://www.practicalmalwareanalysis.com* This string tells us that there is likely some networking functionality within the code, connecting to this domain name
- II. *Microsoft Visual C++ Runtime Library* This tells us the malware code was written in C++.
- III. *‘unable to initialize heap’* It seems as if this malware writer used copious amounts of logging.

4. What happens when you run this malware? Is it what you expected and why?

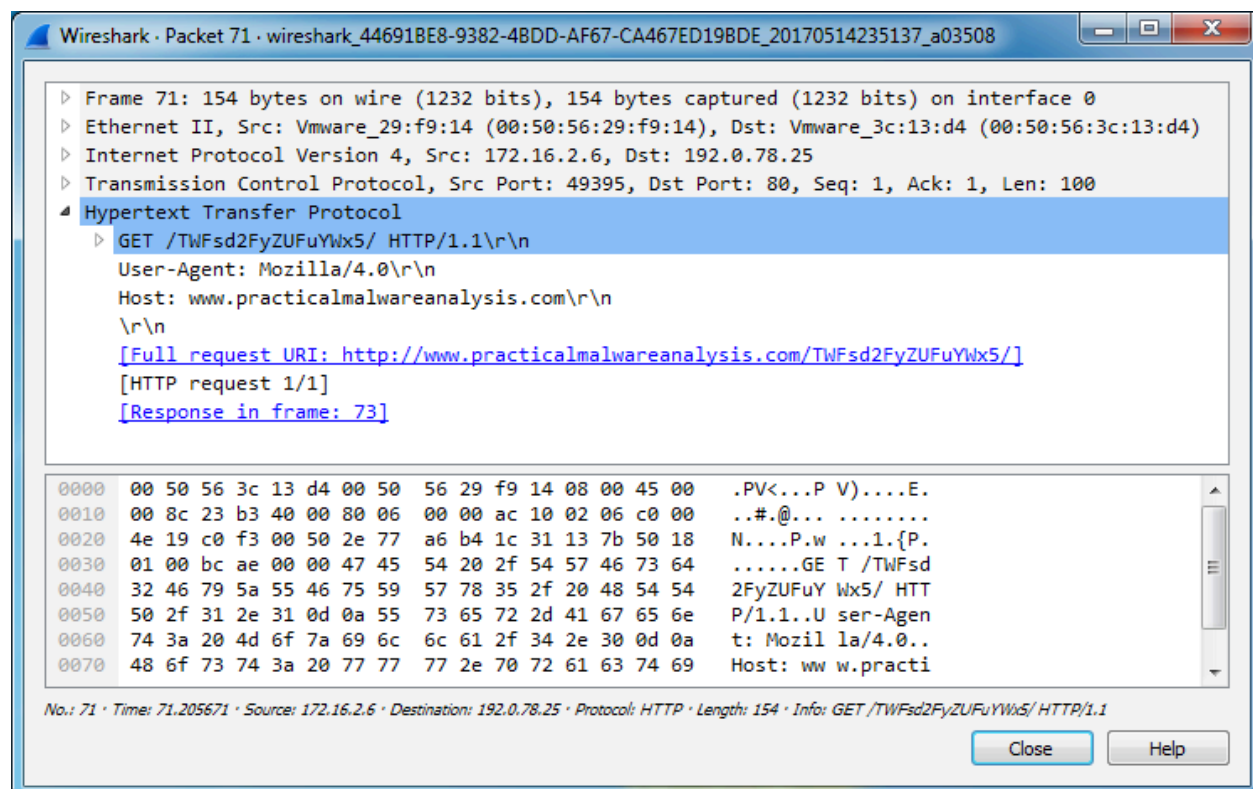
A simple shell opens on the screen. This is not what I expected since usually malware is discreet. In the background, the malware is sending requests and communicating with the host at www.practicalmalwareanalysis.com (IP address is 192.0.78.24).

5. Are there any host based signatures? (Files, registry keys, processes or services, etc.). If so, what are they?

It launches a conhost.exe process, but this is standard for any network-related process connecting outside the host.

6. Are there any network based signatures? (URLs, packet contents, etc.) If so, what are they?

Sending a *HTTP GET /TWfSd2FyZUFuYWx5* request to www.practicalmalwareanalysis.com (IP 192.0.78.24). This packet can be seen below.



7. Is there anything that impeded your analysis? How so? How might you overcome this?

The fact that this malware was packed with UPX makes analyzing it more difficult. It will have to be unpacked from memory using more advanced analysis programs.

8. What do you think is the purpose of the malware?

The malware could potentially be a botnet-creator. Even once the malware has been purged from the machine, the C&C server still pings the machine, indicating the attacker is keeping a record of which IP addresses are infected.