

Practical Malware Analysis — Chapter 1 — Lab 1-2 — Solution



Kamran Saifullah · [Follow](#)

5 min read · Aug 29, 2019



We have recently completed the Lab1-1 questions and found out that it was a

[Open in app](#) ↗

[Sign up](#)

[Sign in](#)



Search



44 / 70 engines detected this file

File Details:

- File Name: Lab01-02.exe
- Size: 3 KB
- Uploaded: 2019-06-10 10:10:25 UTC (2 days ago)
- File Type: EXE

Basic Properties:

Property	Value
MD5	8363436878404da0ae3e46991e355b83
SHA-1	5a016facbcb77e2009a01ea5c67b39af209c3fcb
SHA-256	c876a332d7dd8da331cb8eee7ab7bf32752834d4b2b54eaa362674a2a48f64a6
Authenticating Hash	c0dd97382560a28cc053de86b9505ea78390147de7021744eb49d9b55e3d152f
Imphash	096aa05b6a2e1f2dc66fc73a1a978a7b
SSDEEP	48:atUKzxRhvINZEVtbn4m3ZUJSSeJY8JTalcLoBgs:OUKXktb4KOJzcK
File type	Win32 EXE
Magic	PE32 executable for MS Windows (console) Intel 80386 32-bit
File size	3 KB (3072 bytes)
F-PROT	UPX

History:

Event	Date
Creation Time	2011-01-19 16:10:41
First Seen In The Wild	2010-11-20 23:29:33
First Submission	2011-07-02 17:02:09
Last Submission	2019-05-30 10:41:14
Last Analysis	2019-06-10 10:10:25

Names:

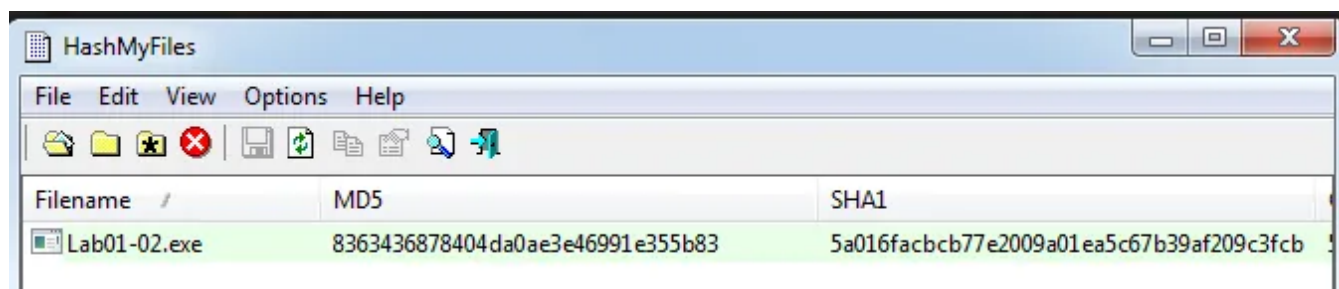
- Lab01-02.exe
- dont_run_me.exe
- Copy of dont_run_me.exe

We can see that this file is malicious. This file was created @ 2011-01-19 16:10:41 and fingerprint is as above!

MD5 8363436878404da0ae3e46991e355b83

SHA-1 5a016facbcb77e2009a01ea5c67b39af209c3fcb

Let's use "HashMyFiles" as it can produce no of hashes for a particular file.



We can see that the hashes are same! Now let's try to check the strings!

```
Strings v2.53 - Search for ANSI and Unicode strings in binary images.  
Copyright (C) 1999-2016 Mark Russinovich  
Sysinternals - www.sysinternals.com  
  
!This program cannot be run in DOS mode.  
Rich  
UPX0  
UPX1  
UPX2  
3.04  
UPX!  
AI3  
h(0  
L$,  
QlI  
" z  
RV$  
u+W  
.hP  
t=p  
sHR  
|Pd  
S`  
a\`Y  
t@E  
DmM  
;0I  
PQ6  
(23h  
MalService  
sHGL345  
http://w  
warean  
ysisbook.co  
om#Int6net Explo!r 8FEI  
.0<  
SystemTimeToFile  
GetMo  
NaA  
Cvg  
*Waitab'r  
Process  
OpenMu$x  
ZSB+  
ForS  
ing  
ObjectU4  
[Vrtb  
CtrlDisp ch  
SCM  
8_e  
Xcpt  
mArg  
sus  
5nm@_  
t_fd
```

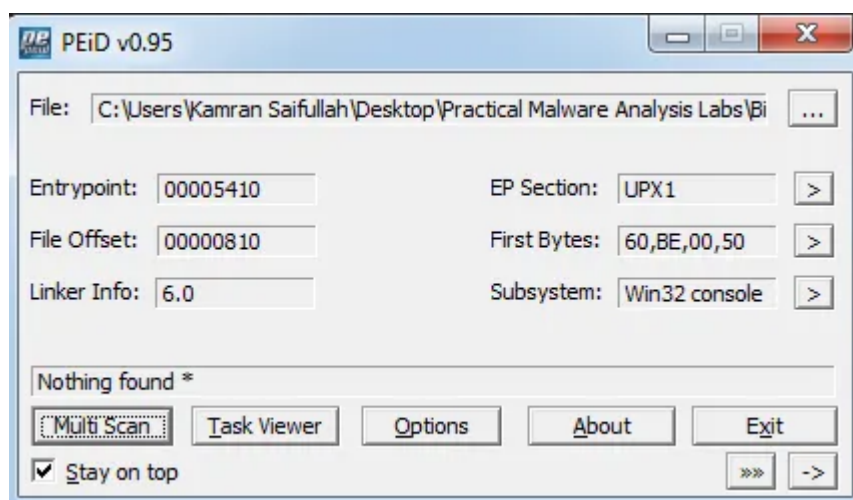
This is the first half and we can clearly see the strings like UPX, UPX0, UPX1, UPX2, UPX3 (UPX is a free and Open-Source Packer for executables) instead of .text, .rdata, .rsrc etc. this probably means that this file is packed. We are able to see URL

(<http://ysisbook.co>), MalService (Mail Service), Internet Explorer 8FEI (Particular Version) being targeted.

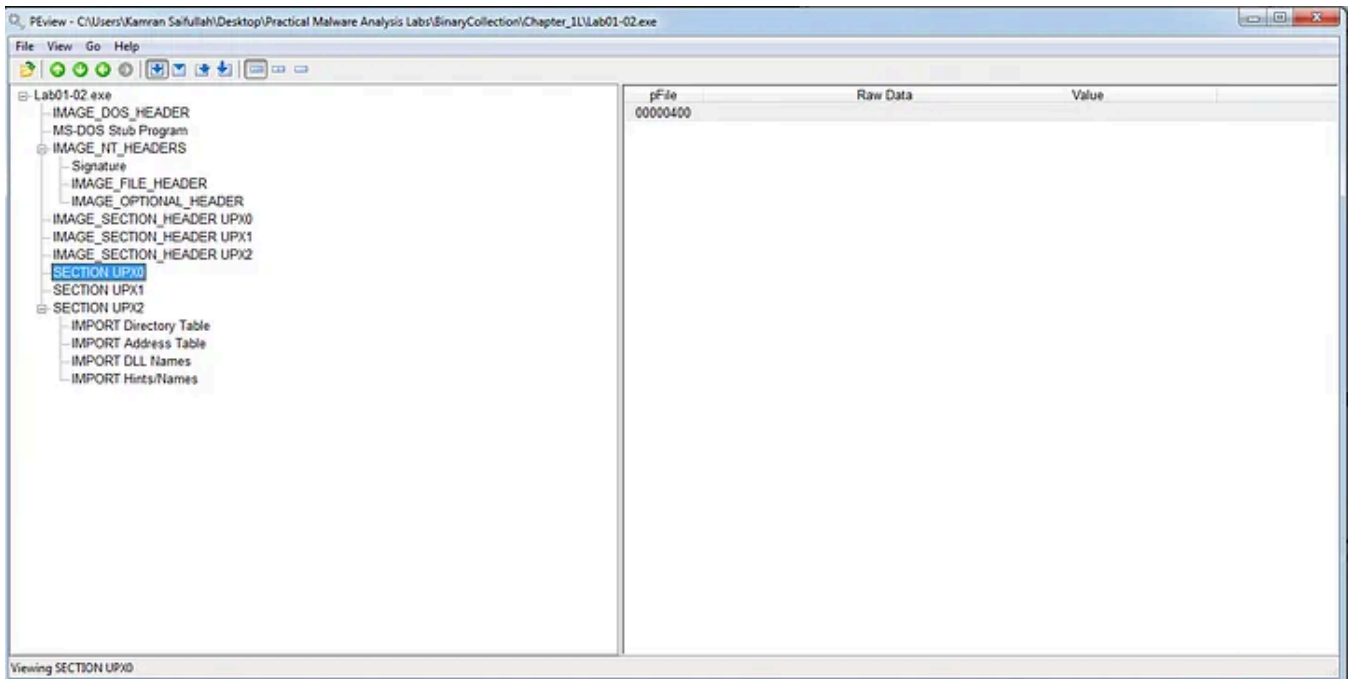
```
i9H
m<e
9,p
vty
d1I37n
o1fp
PEL
dW|6
.4t
1B`.rd
@.&
0'0
_~S
u A
GIu
PTj
XPTPSW
KERNEL32.DLL
ADVAPI32.dll
MSVCRT.dll
WININET.dll
LoadLibraryA
GetProcAddress
VirtualProtect
VirtualAlloc
VirtualFree
ExitProcess
CreateServiceA
exit
InternetOpenA
```

In the second half we can see the DLL files and some of their functionalities being imported like InternetOpenA, CreateServiceA, Virtual*, [GetProcAddress, LoadLibraryA] these two are most commonly seen in packed executable's.

Let's try running the PEiD on this executable!



We can see that this executable has been packed using UPX1 as found in EP Section. Now we can analyze the PE Header using PView and we can clearly see that the sections have been packed/obfuscated.



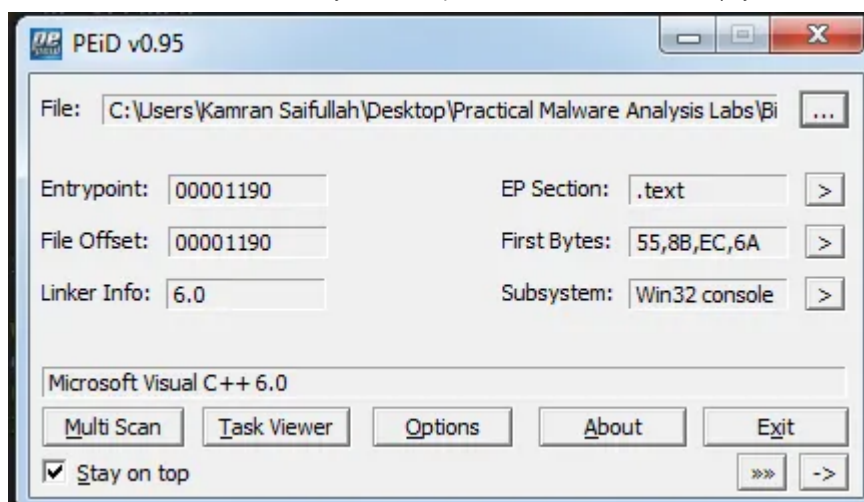
Now we need to unpack this executable. This can be done by using the PE-Explorer plugins and also you can download the UPX tool from github.

```
C:\Users\Kamran Saifullah\Desktop\Practical Malware Analysis Labs\BinaryCollection\Chapter_1
λ upx.exe -o Lab01-02_unpacked.exe -d Lab01-02.exe
Ultimate Packer for eXecutables
Copyright (C) 1996 - 2018
UPX 3.95w Markus Oberhumer, Laszlo Molnar & John Reiser Aug 26th 2018

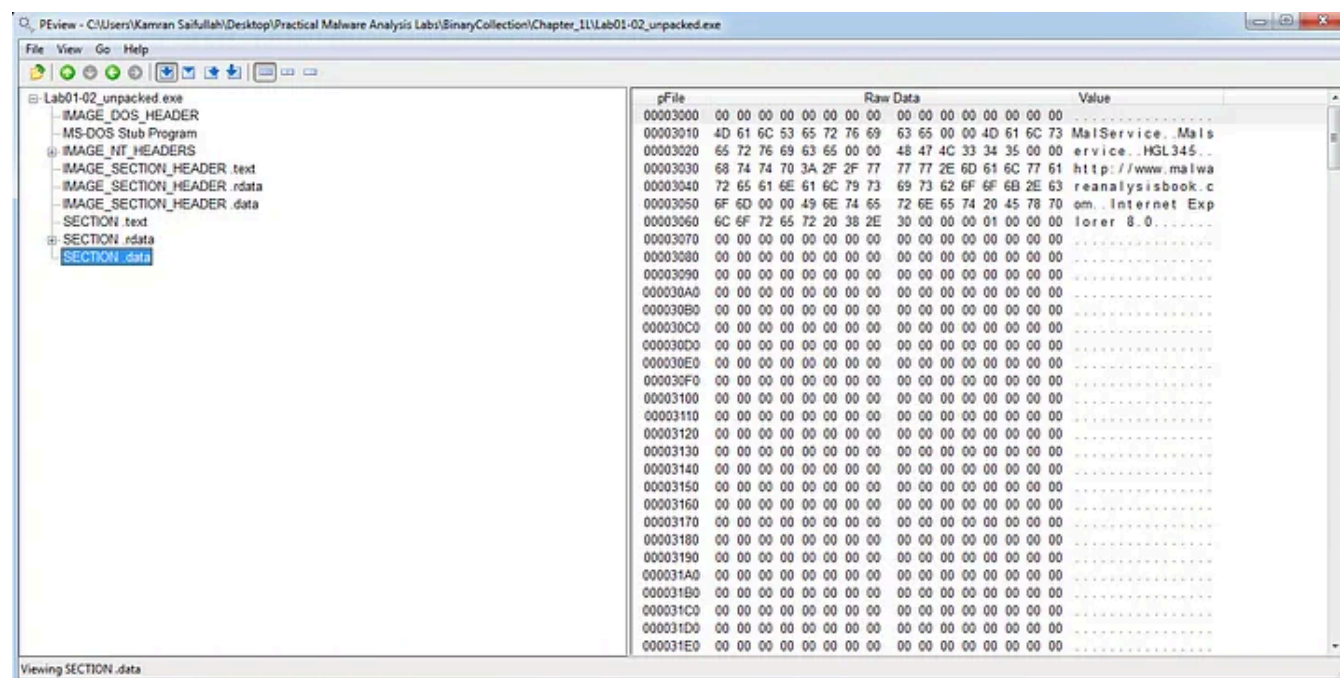
File size      Ratio      Format      Name
-----
16384 <-      3072      18.75%      win32/pe      Lab01-02_unpacked.exe

Unpacked 1 file.
```

Let's run the PEiD on this file now!



We have successfully unpacked the executable. Now we can run the PView to look for the sections!



We are able to see the PE Header sections now and the data they contain. At this moment we can run the strings command to check the strings and also DependencyWalker to check the DLL and their corresponding functionalities which are being imported.

On running the strings we are able to see much much more data!


```

KERNEL32.DLL
ADVAPI32.dll
MSVCRT.dll
WININET.dll
SystemTimeToFileTime
GetModuleFileNameA
CreateWaitableTimerA
ExitProcess
OpenMutexA
SetWaitableTimer
WaitForSingleObject
CreateMutexA
CreateThread
CreateServiceA
StartServiceCtrlDispatcherA
OpenSCManagerA
_exit
_XcptFilter
exit
__p__initenv
__getmainargs
__initterm
__setusermatherr
__adjust_fdiv
__p__commode
__p__fmode
__set_app_type
_except_handler3
_controlfp
InternetOpenUrlA
InternetOpenA
MalService
MalService
HGL345
http://www.malwareanalysisbook.com
Internet Explorer 8.0

C:\Users\Kamran Saifullah\Desktop\Practical Malware Analysis Labs\BinaryCollection\Chapter_1L
λ

```

Now we are getting the clearer view of what this executable might do when executed!

On running the DependencyWalker on this executable we can see that it is importing functions from 4 DLLs.

PI	Ordinal ^	Hint	Function	Entry Point
	N/A	0 (0x0000)	InternetOpenUrlA	Not Bound
	N/A	0 (0x0000)	InternetOpenA	Not Bound

E	Ordinal ^	Hint	Function	Entry Point
001	(0x0065)	N/A	N/A	0x000145BF
002	(0x0066)	N/A	N/A	0x00032E25
003	(0x0067)	N/A	N/A	0x0009C836
004	(0x0068)	N/A	N/A	0x00091425
005	(0x0069)	N/A	N/A	0x00091779
006	(0x006A)	17 (0x0011)	DispatchAPICall	0x0007F011
007	(0x006B)	0 (0x0000)	CommitUICacheEntryA	0x00048665
008	(0x006C)	N/A	N/A	0x00077A45

This executable will be connecting to the <http://malwareanalysisbook.com> and will run it under the name of MalService.

Now that is enough static analysis on this LAB. Let's answer the questions!

Lab 1-2

Analyze the file Lab02-02.exe

Questions

1. Upload the Lab01-02.exe file to <http://www.VirusTotal.com/>. Does it match any existing antivirus definitions?

We have uploaded the file and have found that it matched the existing antivirus definitions.

2. Are there any indications that this file is packed or obfuscated? If so, what are these indicators? If the file is packed, unpack it if possible?

We found out that this executable was packed and we were also able to unpack it using the UPX tool.

3. Do any imports hint at this program's functionality? If so, which imports are they and what do they tell you?

a. InternetOpenA → Initializes an application's use of the WinINet functions we can see what user agent is used to initiate the connection.

b. InternetOpenUrlA → Opens a FTP or HTTP URL

c. CreateMutexA → Create mutex lock to prevent multiple running instances of the malware

d. OpenMutexA → Open a created mutex

e. CreateServiceA → Create a service object to the victim's machine. Often use for persistence.

f. OpenSCManagerA → called before CreateService is invoked to establish a connection to the service control manager

g. StartServiceCtrlDispatcherA → When the service control manager starts a service process, it waits for the process to call the **StartServiceCtrlDispatcher** function. The main thread of a service process should make this call as soon as possible after it starts up (within 30 seconds)

As far as i have understood from the details. This executable is connecting to a website and then binding it as a service MalService for persistence. It's somehow going to be a part of C&C system. Receiving the commands from the web and then executing them on the machine!

4. What host- or network-based indicators could be used to identify this malware on infected machines?

We can look for the service named MalService via services.msc also we can check for the dnslookups for <http://malwareanalysisbook.com/> via a specific Internet Explorer string which will be passed via the browser user-agent FIELD. Moreover we can create a firewall rule to block such traffic.

Security



Follow

Written by Kamran Saifullah

378 Followers

Malware/RE/Firmware Analysis, App Sec/Off Sec, VAPT, Phishing Simulations/SE | Risk Management, IS Governance, Audits, ISO 27001 LI

More from Kamran Saifullah

```

.000000..0 080          0000          .000000.          .0          .00
00.
d8P'  `Y8  `"'          `888          d8P'  `Y8b          0888          .dP'"
Y88b
Y88bo.          0000  .00000.  888  0000  888          888  .0000.0  888
]8P'
`"Y88880.  `888  d88'  `"Y8  888  .8P'  888          888  d88(  "8  888          .
d8P'
`"Y88b  888  888          888888.  888          888  `"Y88b.  888          .dP
,
oo  .d8P  888  888  .o8  888  `88b.  `88b  d88'  o.  )88b  888  .o.  .oP
.o
8""88888P'  08880  `Y8bod8P'  08880  08880  `Y8bood8P'  8""888P'  08880  Y8P  88888
88888

```

By @D4rk36

ubuntu login: _

 Kamran Saifullah

SickOS 1.2 WalkThrough

Hi,

5 min read · Apr 14, 2018

 10 


DISCLAIMER!

We at Kioptrix are not responsible for any damaged directly, or indirectly, caused by using this system. We suggest you do not connect this installation to the Internet. It is, after all, a vulnerable setup. Please keep this in mind when playing the game.

This machine is setup to use DHCP.
Before playing the game, please modify your attacker's hosts file.
<ip> kioptrix3.com
This challenge contains a Web Application.

If you have any questions, please direct them to:
comms[at]kioptrix.com
Hope you enjoy this challenge.
-Kioptrix Team

Ubuntu 8.04.3 LTS Kioptrix3 tty1

Kioptrix3 login: _

 Kamran Saifullah

WalkThrough! Kioptrix — 3 By VulnHub

Hi,

8 min read · Mar 13, 2018



24



52

Written by: Jeremy Hui

Keith is watching chickens cross a road in his grandfather's farm. He once heard from his grandfather that there was something significant about this behavior, but he can't figure out why. Help Keith discover what the chickens are doing from this seemingly simple behavior.

[hsctf-chicke...](#)

Kamran Saifullah

HSCTF 6—Forensics Challenges—Solutions

After publishing the solutions of the web challenges now it's time to move on with forensics challenges and this is all about how i solved...

5 min read · Jun 13, 2019



17



```
untu 12.04.4 LTS SickOs tty1  
ckOs login: _
```



Kamran Saifullah

SickOS 1.1 Walkthrough

Hi,

5 min read · Apr 11, 2018

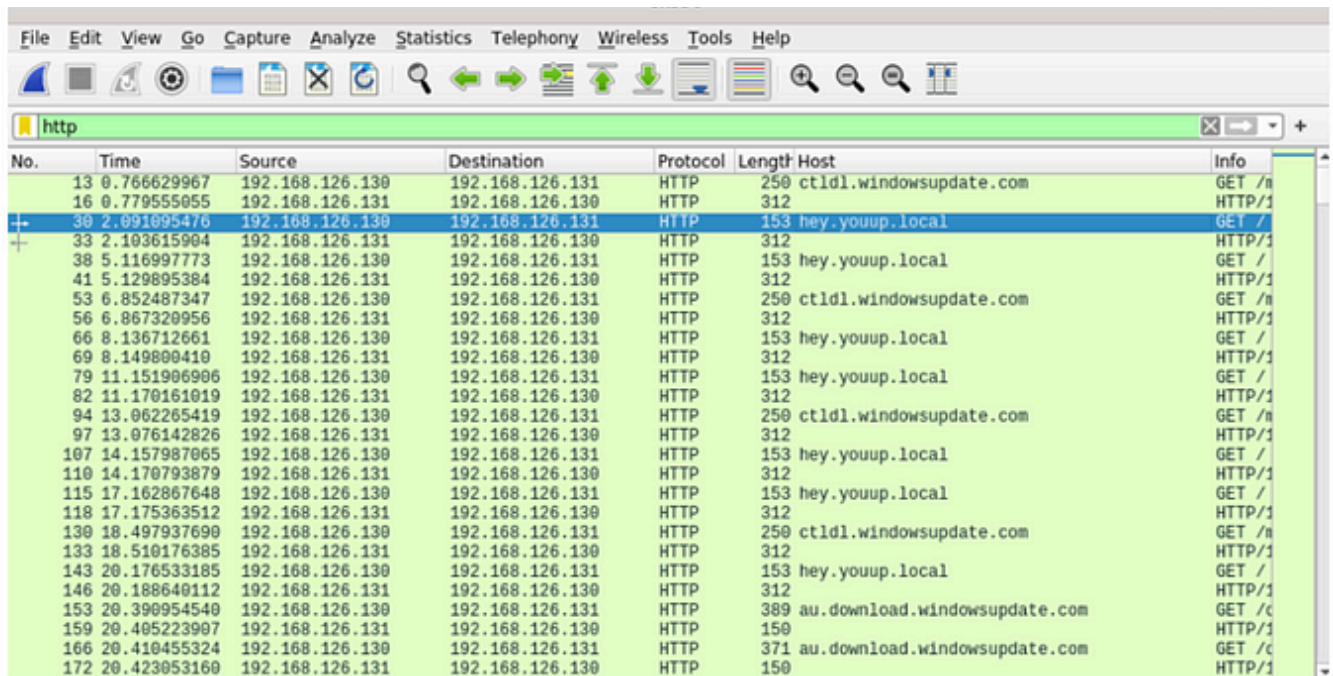


9



See all from Kamran Saifullah

Recommended from Medium



No.	Time	Source	Destination	Protocol	Length	Host	Info
13	0.766629967	192.168.126.130	192.168.126.131	HTTP	250	ctld1.windowsupdate.com	GET /n
16	0.779555055	192.168.126.131	192.168.126.130	HTTP	312		HTTP/1
30	2.091095476	192.168.126.130	192.168.126.131	HTTP	153	hey.youup.local	GET /
33	2.103615904	192.168.126.131	192.168.126.130	HTTP	312		HTTP/1
38	5.116997773	192.168.126.130	192.168.126.131	HTTP	153	hey.youup.local	GET /
41	5.129895384	192.168.126.131	192.168.126.130	HTTP	312		HTTP/1
53	6.852487347	192.168.126.130	192.168.126.131	HTTP	250	ctld1.windowsupdate.com	GET /n
56	6.867320956	192.168.126.131	192.168.126.130	HTTP	312		HTTP/1
66	8.136712661	192.168.126.130	192.168.126.131	HTTP	153	hey.youup.local	GET /
69	8.149800410	192.168.126.131	192.168.126.130	HTTP	312		HTTP/1
79	11.151906906	192.168.126.130	192.168.126.131	HTTP	153	hey.youup.local	GET /
82	11.170161019	192.168.126.131	192.168.126.130	HTTP	312		HTTP/1
94	13.062265419	192.168.126.130	192.168.126.131	HTTP	250	ctld1.windowsupdate.com	GET /n
97	13.076142826	192.168.126.131	192.168.126.130	HTTP	312		HTTP/1
107	14.157987065	192.168.126.130	192.168.126.131	HTTP	153	hey.youup.local	GET /
110	14.170793879	192.168.126.131	192.168.126.130	HTTP	312		HTTP/1
115	17.162867648	192.168.126.130	192.168.126.131	HTTP	153	hey.youup.local	GET /
118	17.175363512	192.168.126.131	192.168.126.130	HTTP	312		HTTP/1
130	18.497937690	192.168.126.130	192.168.126.131	HTTP	250	ctld1.windowsupdate.com	GET /n
133	18.510176385	192.168.126.131	192.168.126.130	HTTP	312		HTTP/1
143	20.176533185	192.168.126.130	192.168.126.131	HTTP	153	hey.youup.local	GET /
146	20.188640112	192.168.126.131	192.168.126.130	HTTP	312		HTTP/1
153	20.390954540	192.168.126.130	192.168.126.131	HTTP	389	au.download.windowsupdate.com	GET /c
159	20.405223907	192.168.126.131	192.168.126.130	HTTP	150		HTTP/1
166	20.410455324	192.168.126.130	192.168.126.131	HTTP	371	au.download.windowsupdate.com	GET /c
172	20.423053160	192.168.126.131	192.168.126.130	HTTP	150		HTTP/1

 Hüseyin EKŞİ

Malware Analysis of PMAT-Bonus Unknown malware

I have analyzed the Bonus malware called unknown and would like to share my findings. If you have analyzed this piece of malware please...

3 min read · Jan 21, 2024



13



Abdelwahab Shandy

CyberDefenders :Qradar101 Blue Team Challenge

Category : Threat Hunting

11 min read · Nov 23, 2023



4



Lists



Staff Picks

630 stories · 920 saves



Stories to Help You Level-Up at Work

19 stories · 581 saves



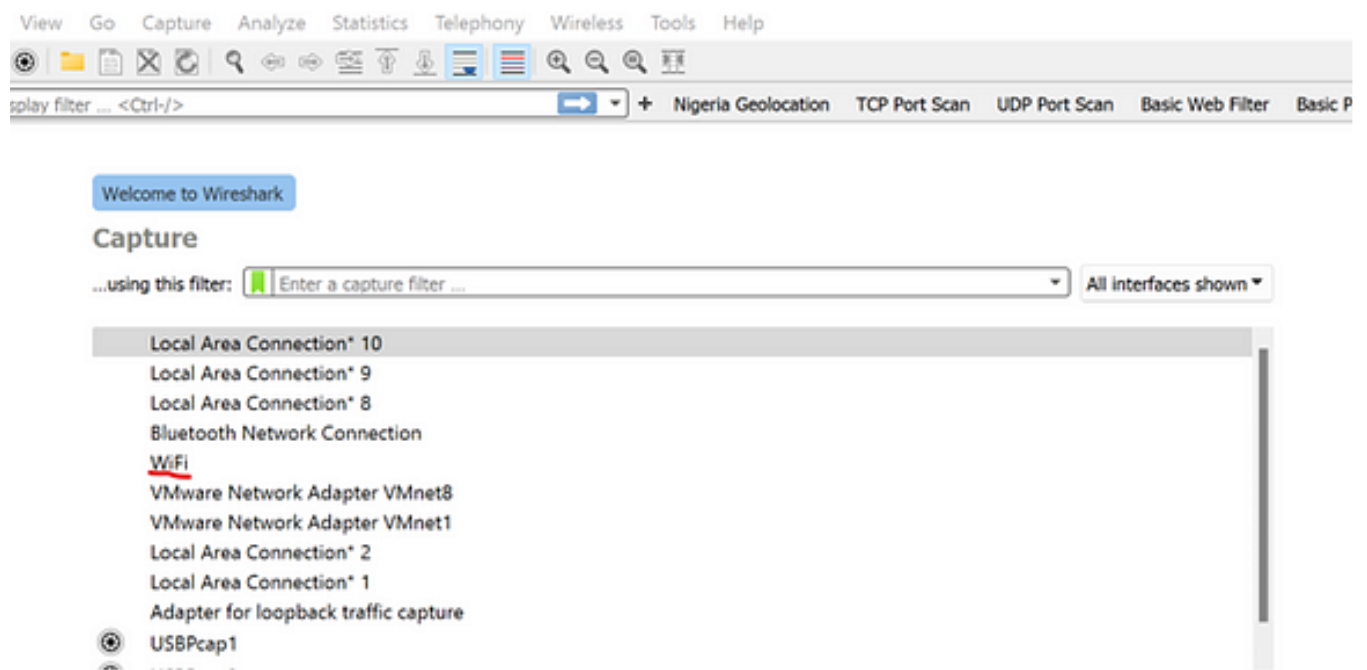
Self-Improvement 101

20 stories · 1675 saves



Productivity 101

20 stories · 1543 saves




Kevin Moore

Analyzing PCAP Files using Wireshark

Hello, thanks for stopping by to read this blog. Its a deep-dive into the use of Wireshark to investigate captured network traffic.

6 min read · Nov 20, 2023



 Rewa Aslekar

Malware Analysis

A Beginner's guide to Malware Analysis

13 min read · Oct 31, 2023





Yen

TryHackMe Writeups—Basic Malware RE

Are you ready to dive in the field of Malware Analysis?

3 min read · Nov 11, 2023



Worldsleaks

OpenWire—

CyberDefenders CTF

7 min read · Jan 25, 2024



13



See more recommendations