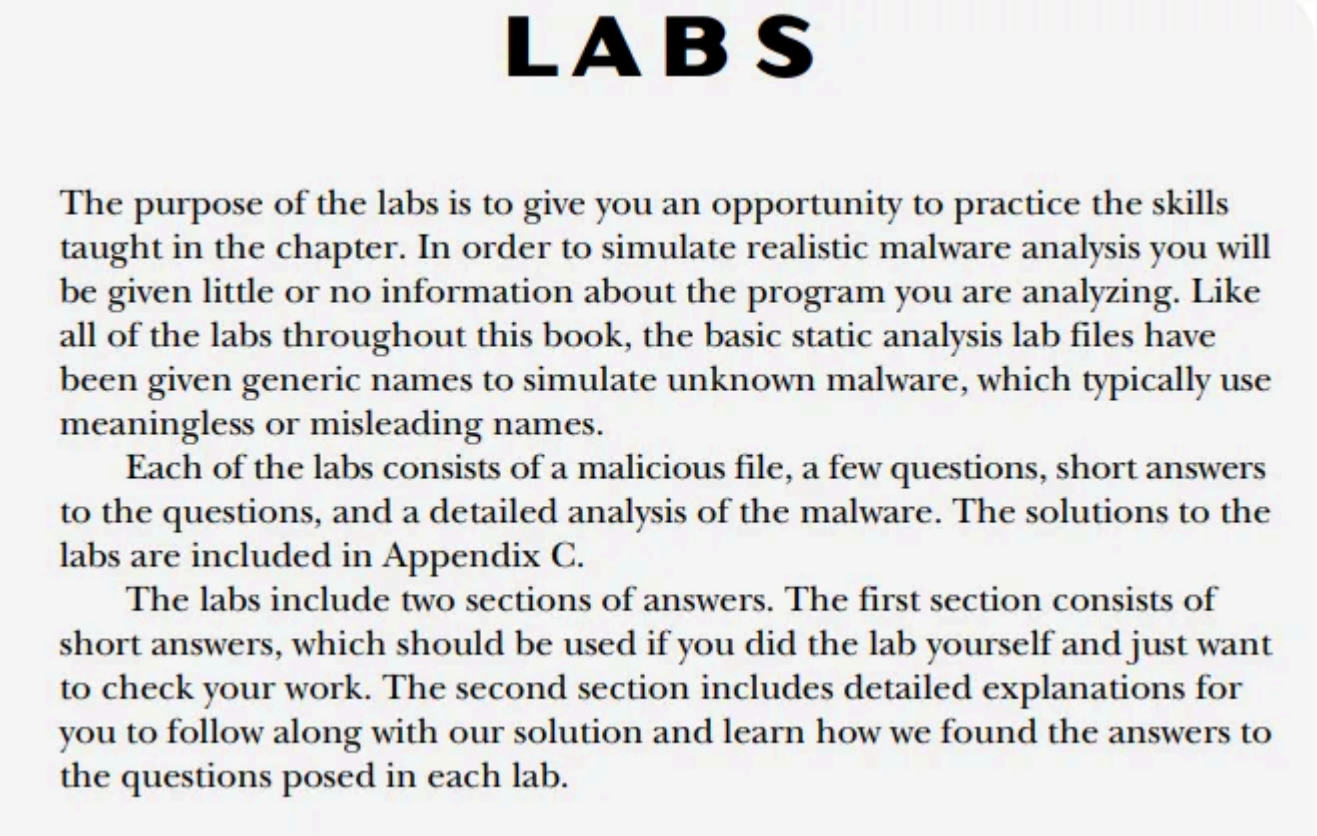


LABS

The purpose of the labs is to give you an opportunity to practice the skills taught in the chapter. In order to simulate realistic malware analysis you will be given little or no information about the program you are analyzing. Like all of the labs throughout this book, the basic static analysis lab files have been given generic names to simulate unknown malware, which typically use meaningless or misleading names.

Each of the labs consists of a malicious file, a few questions, short answers to the questions, and a detailed analysis of the malware. The solutions to the labs are included in Appendix C.

The labs include two sections of answers. The first section consists of short answers, which should be used if you did the lab yourself and just want to check your work. The second section includes detailed explanations for you to follow along with our solution and learn how we found the answers to the questions posed in each lab.

Practical Malware Analysis — Book

Practical Malware Analysis — Chapter 1 — Labs 1–1 — Solution

Kamran Saifullah · Follow

4 min read · Aug 29, 2019

As we are done with the Chapter-1. It's time to work on the labs to get most out of our learning. So let's begin.

> *Note:* I have copied the Labs Details (text) from the book as it is.

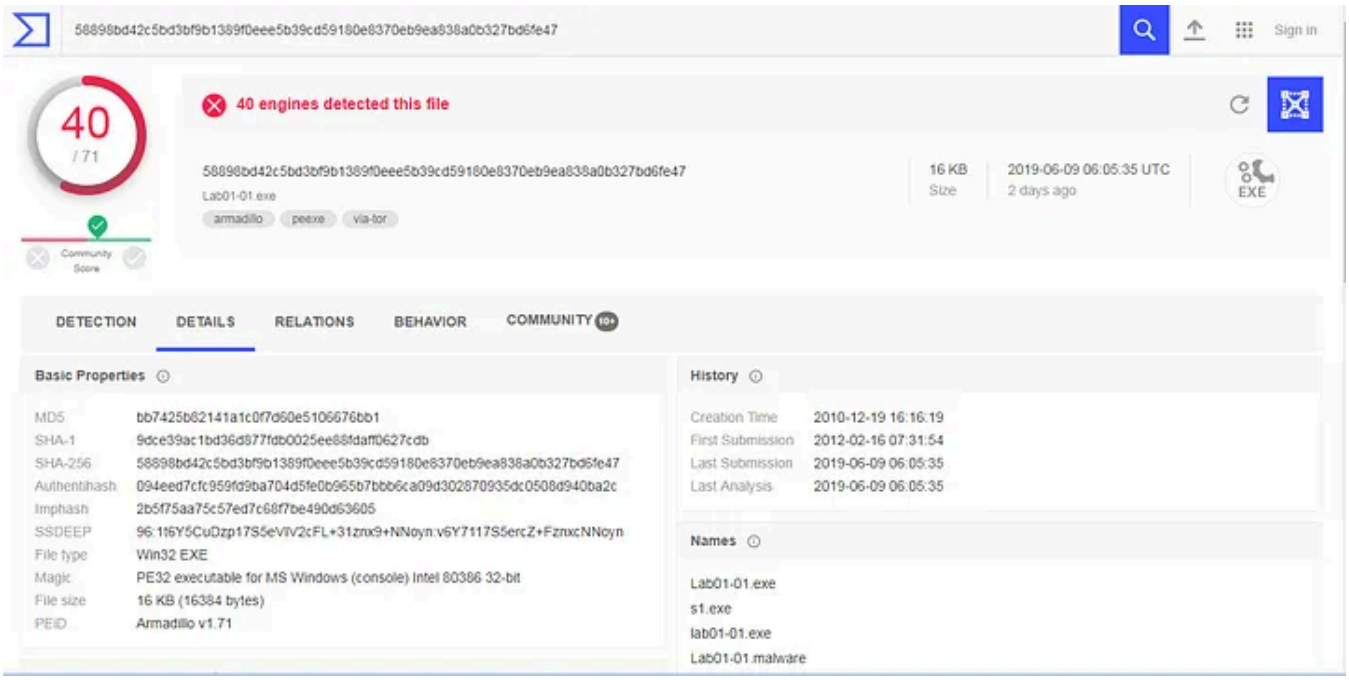
Lab 1–1

This lab uses the files Lab01–01.exe and Lab01–01.dll. Use the tools and techniques described in the chapter to gain information about the files and answer the questions below.

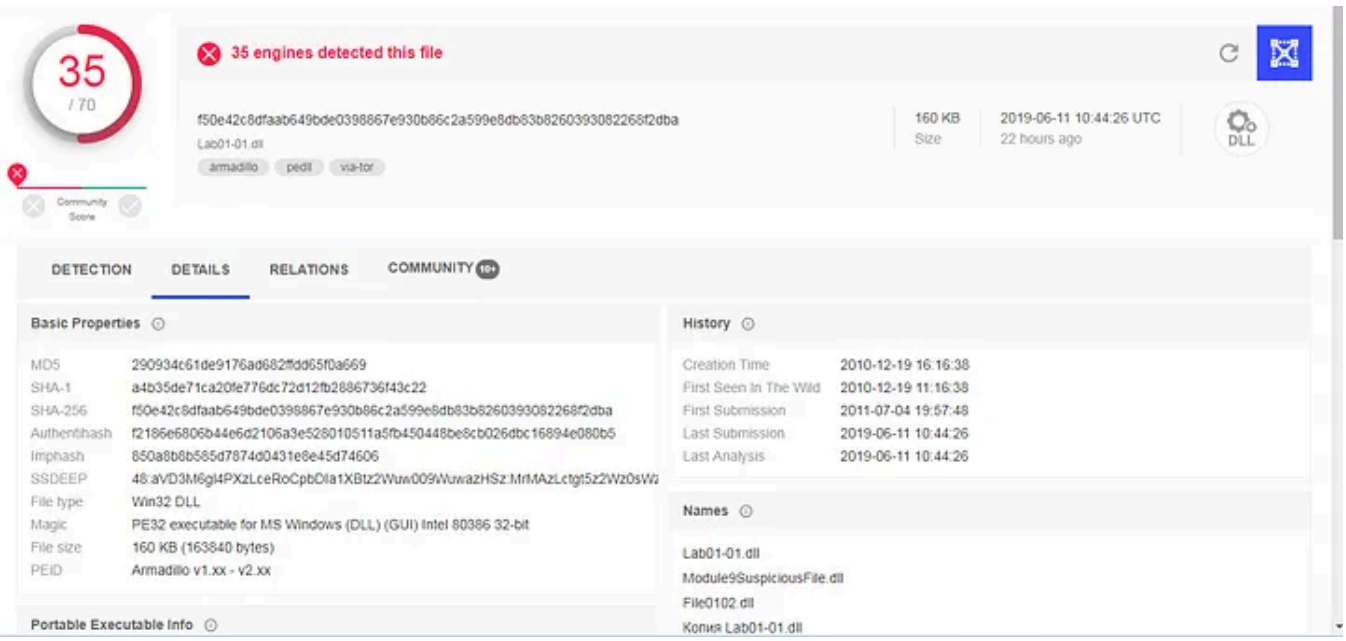
Questions

1. **Upload the files to http://www.VirusTotal.com/ and view the reports. Does either file match any existing antivirus signatures?**

Let's upload both of the files on VirusTotal and tally the result!



Result of Lab01–01.exe



Result of — Lab01–01.dll

We can clearly see that these files have been matched with the previously known signatures and have also been detected as malicious.

2. **When were these files compiled?**

The compilation time of both file as per the report of VirusTotal is.

Lab01–01.exe → 2010–12–19 16:16:19

Lab01–01.dll → 2010–12–19 16:16:38

We can also find the compilation time using PEview and checking the IMAGE_FILE_HEADER details.

3. **Are there any indications that either of these files is packed or obfuscated? If so, what are these indicators?**

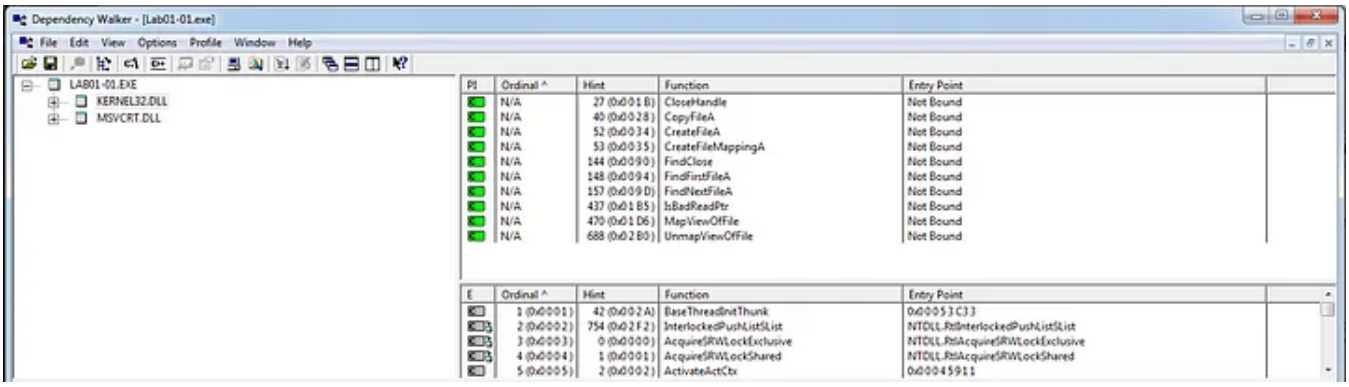
PEiD can be used to find the packed or obfuscated file although we were able to find all the necessary details and the strings. So we conclude that both of these files were not been packed or obfuscated.

4. **Do any imports hint at what this malware does? If so, which imports are they?**

Both, the executable and DLL file do imports. **Lab01–01.exe** does the following imports.

```
λ strings Lab01-01.exe | grep dll | uniq -u
KERNEL32.dll
MSVCRT.dll
kerne132.dll
kernel32.dll
C:\windows\system32\kerne132.dll
Lab01-01.dll
C:\Windows\System32\Kernel32.dll
```

a. KERNEL32.dll — Following functions from this library were called.



CreateFileA → Creates or opens a file or I/O device.

CopyFileA → Copies an existing file to a new file.

CreateFileMappingA → Creates or opens a named or unnamed file mapping object for a specified file.

FindFirstFileA → Searches a directory for a file or subdirectory with a name that matches a specific name (or partial name if wildcards are used).

FindNextFileA → Continues a file search from a previous call.

MapViewOfFile → Maps a view of a file mapping into the address space of a calling process. Malware can make changes to the actual file once it is mapped.

b. **MSVCRT.dll** → A module containing standard C library functions such as printf, memcpy, and cos. It is a part of the Microsoft C Runtime Library. Non-system processes like **msvcrt.dll** originate from software you installed on your system.

c. **kernel132.dll** → Disguised version of original KERNEL32.DLL.

d. **Lab01–01.dll** → Additional DLL file created for the successful working of Lab01–01.exe executable.

The second file **Lab01–01.dll** do the following imports.

```
C:\Users\Kamran Saifullah\Desktop\Practical M
λ strings Lab01-01.dll | grep dll | uniq -u
KERNEL32.dll
WS2_32.dll
MSVCRT.dll
```

KERNEL32.dll → **Kernel32.dll** is the 32-bit dynamic link library found in the Windows operating system kernel. It handles memory management, input/output operations, and interrupts. When Windows boots up, **kernel32.dll** is loaded into a protected memory space so other applications do not take that space over.

MSVCRT → A module containing standard C library functions such as printf, memcpy, and cos. It is a part of the Microsoft C Runtime Library. Non-system processes like **msvcrt.dll** originate from software you installed on your system.

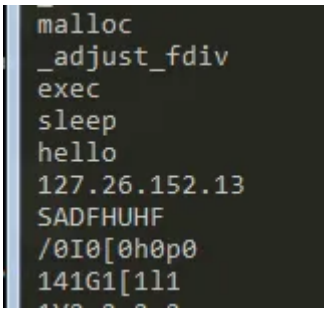
WS2_32.dll → The Windows Sockets Library **ws2_32.dll**, is required by windows and applications to handle network connections.

5. **Are there any other files or host-based indicators that you could look for on infected systems?**

While finding the strings we found that there is another file named as "Kerne132.dll" which is supposed to be disguised as the "Kernel32.dll". Also there is another "Lab01–01.DLL" which is not a common OS DLL. So we can look for these files on the system.

6. **What network-based indicators could be used to find this malware on infected machines?**

We found an IP address when we checked the string. So we capture all the network traffic from all the systems and can look for the communication that is being done over this IP address.



7. **What would you guess is the purpose of these files?**

On bringing up all the pieces together we can assume that Lab01–01.exe along with the extension Lab01–01.dll is a malware which creates a backdoor and connects to a C&C server and transfer the critical information. Secondly both of the files are not

packed and Lab01–01.exe searches in and from directories and look for a particular files and replaces them with disguised files. Also it imports functions from core KERNEL32.DLL and network based imports to establish the connections. Also uses the exec function which means that it would be executing some other programs/files along with sleep function which waits until a particular statement or piece of code gets executed. This is mostly used in backdoors.

Security

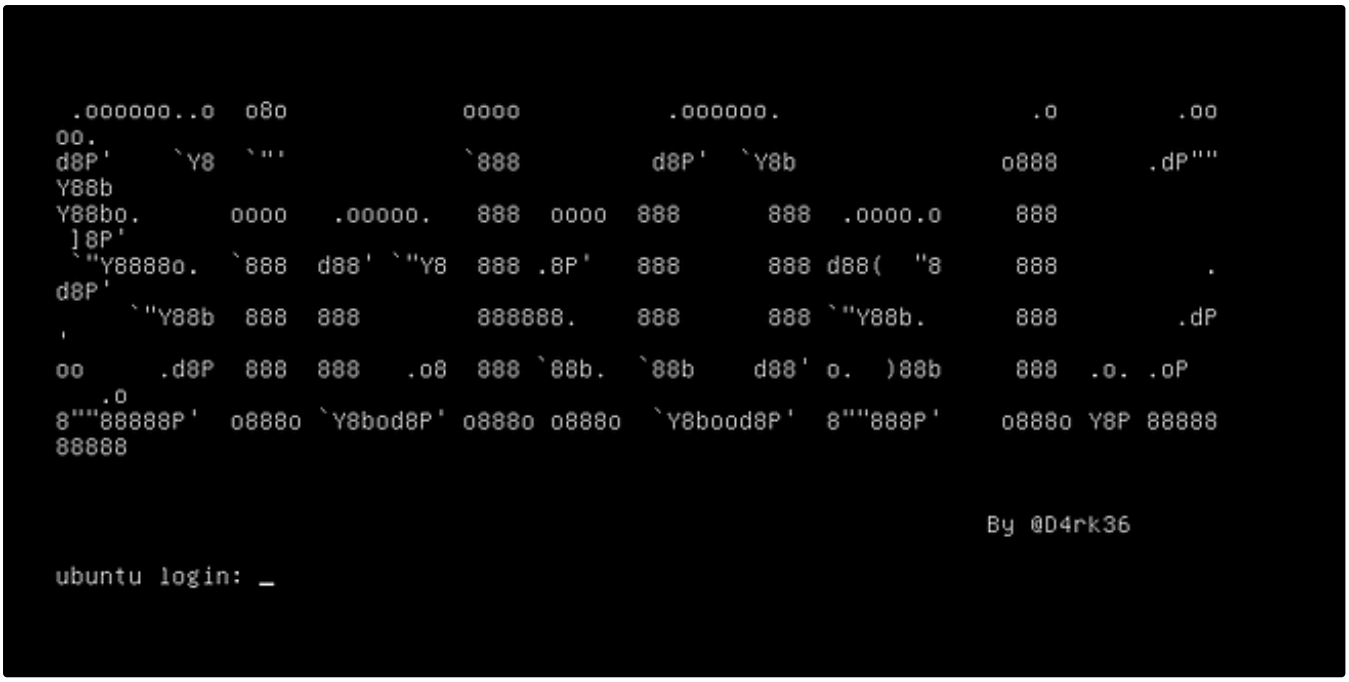
Follow

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



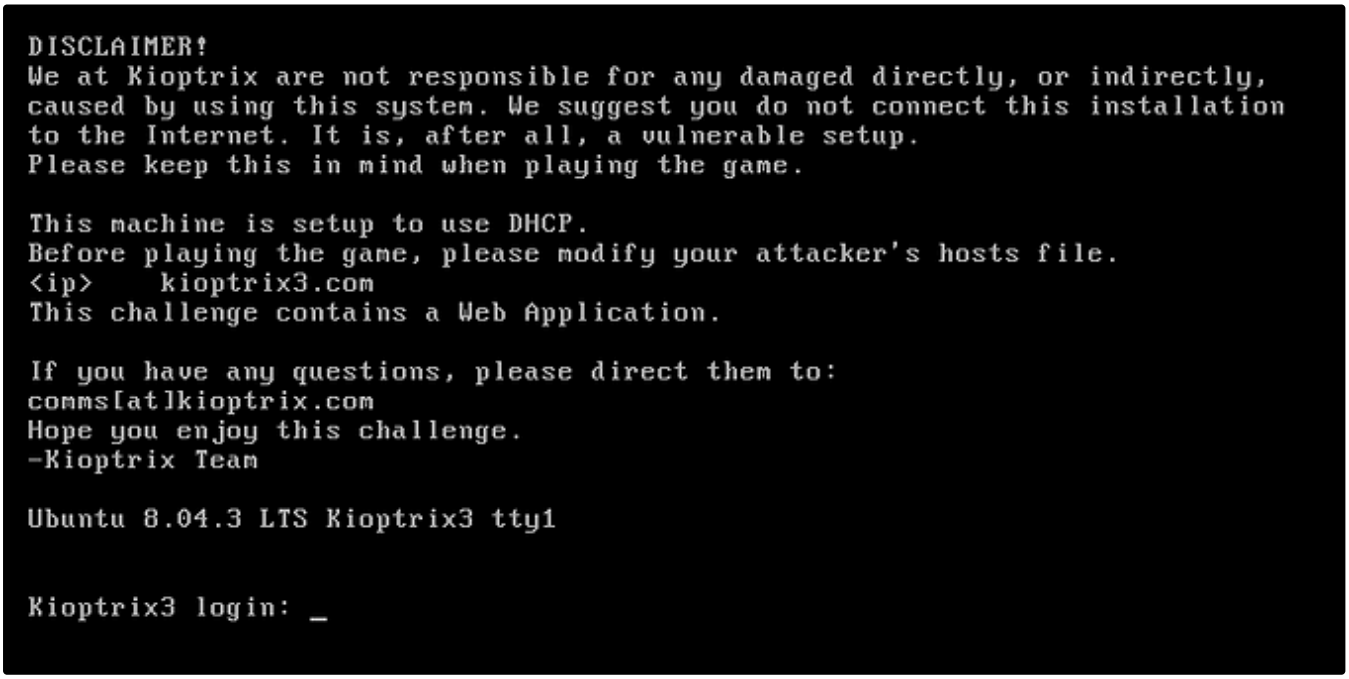
Kamran Saifullah

SickOS 1.2 WalkThrough

Hi,

5 min read · Apr 14, 2018

10



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...

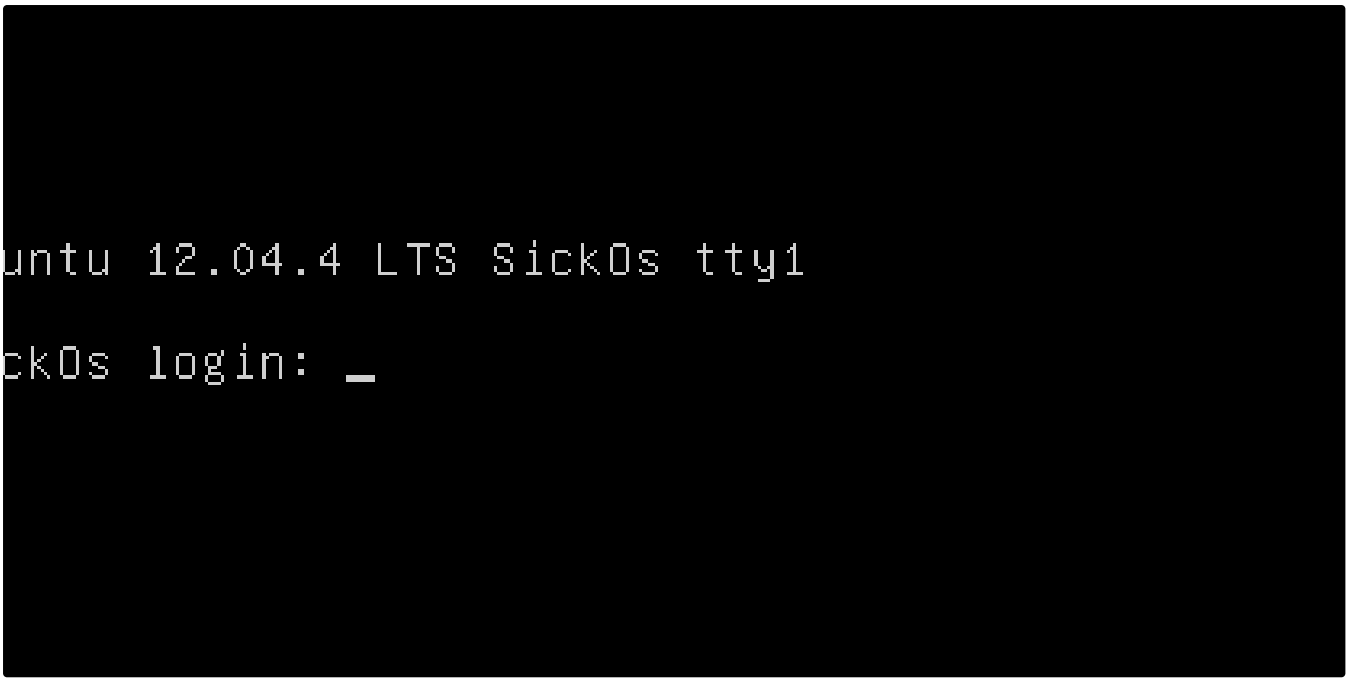
Kamran Saifullah

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



Kamran Saifullah

SickOS 1.1 Walkthrough

Hi,

5 min read · Apr 11, 2018

9

See all from Kamran Saifullah

Recommended from Medium

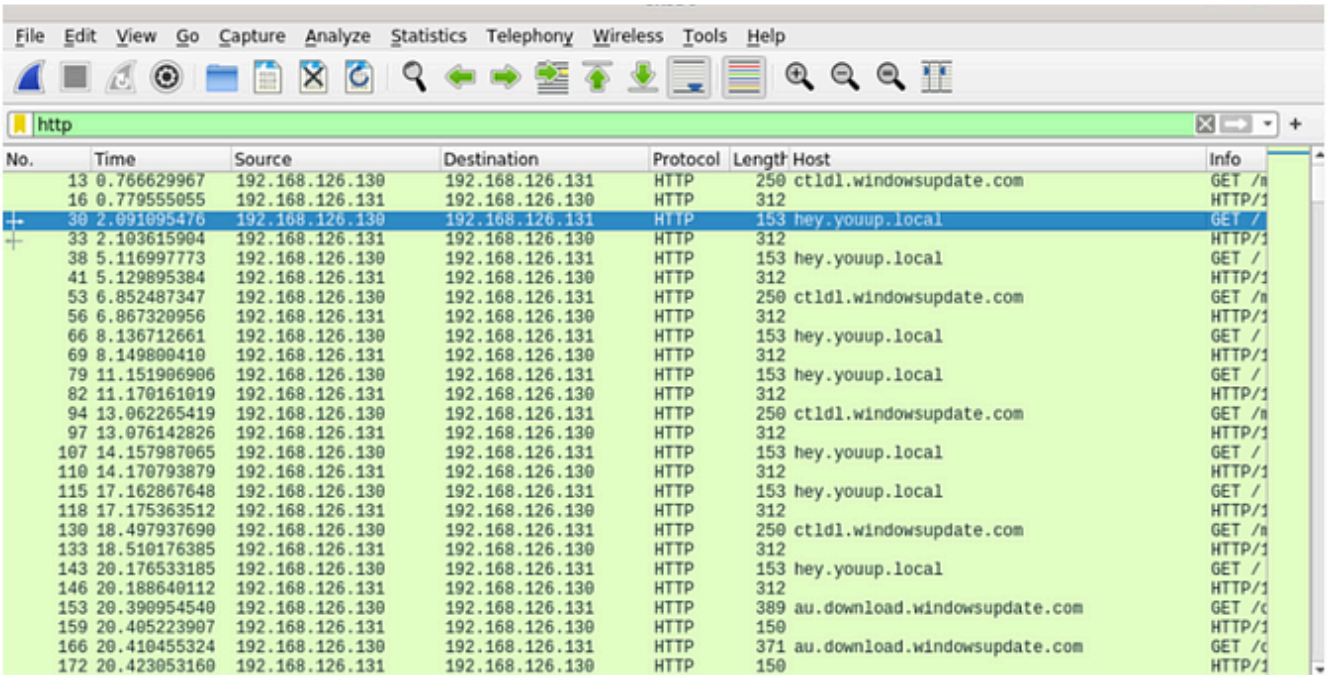
Abdelwahab Shandy

CyberDefenders :Qradar101 Blue Team Challenge

Category : Threat Hunting

11 min read · Nov 23, 2023

4



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

Lists

Staff Picks

630 stories · 920 saves

Stories to Help You Level-Up at Work

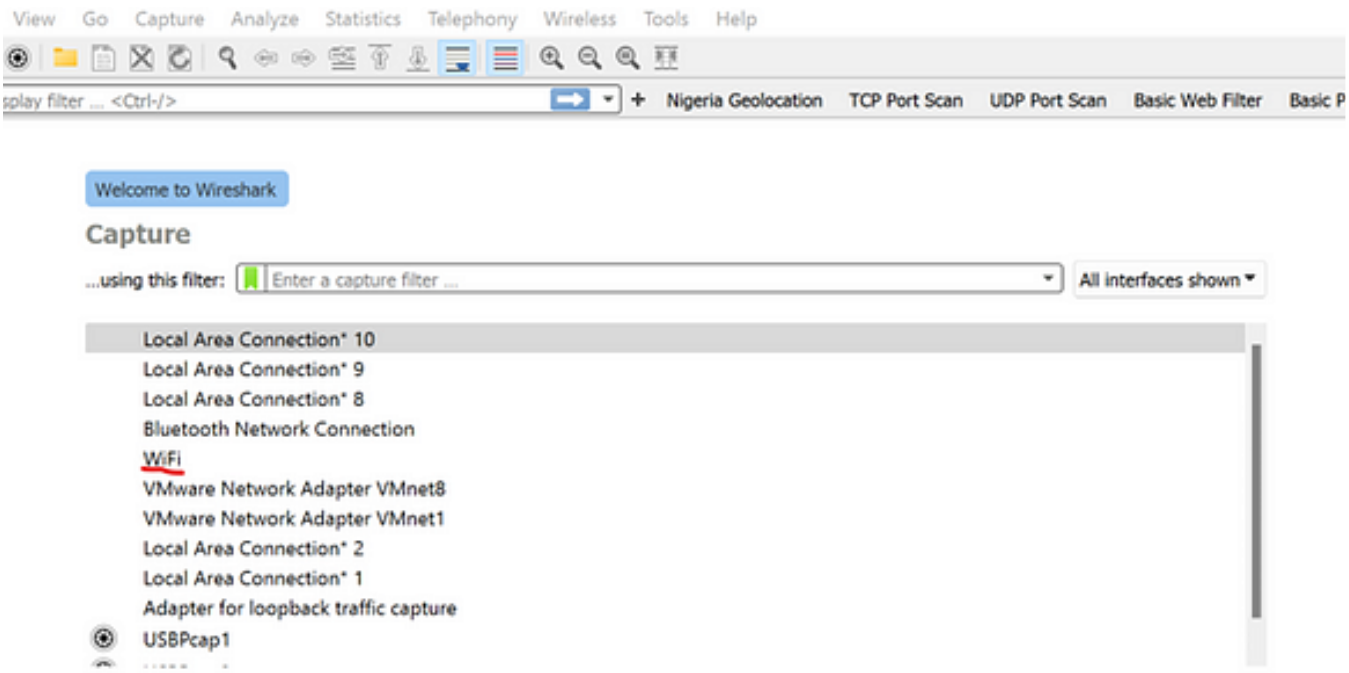
19 stories · 581 saves

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

7

Zaid Khaishagi

GHIDRA TUTORIAL: CTF CHALLENGE

This is the final article in the Ghidra Tutorial series. In this tutorial, we will make use of Ghidra to solve an actual CTF challenge. We...

8 min read · Jan 2, 2024

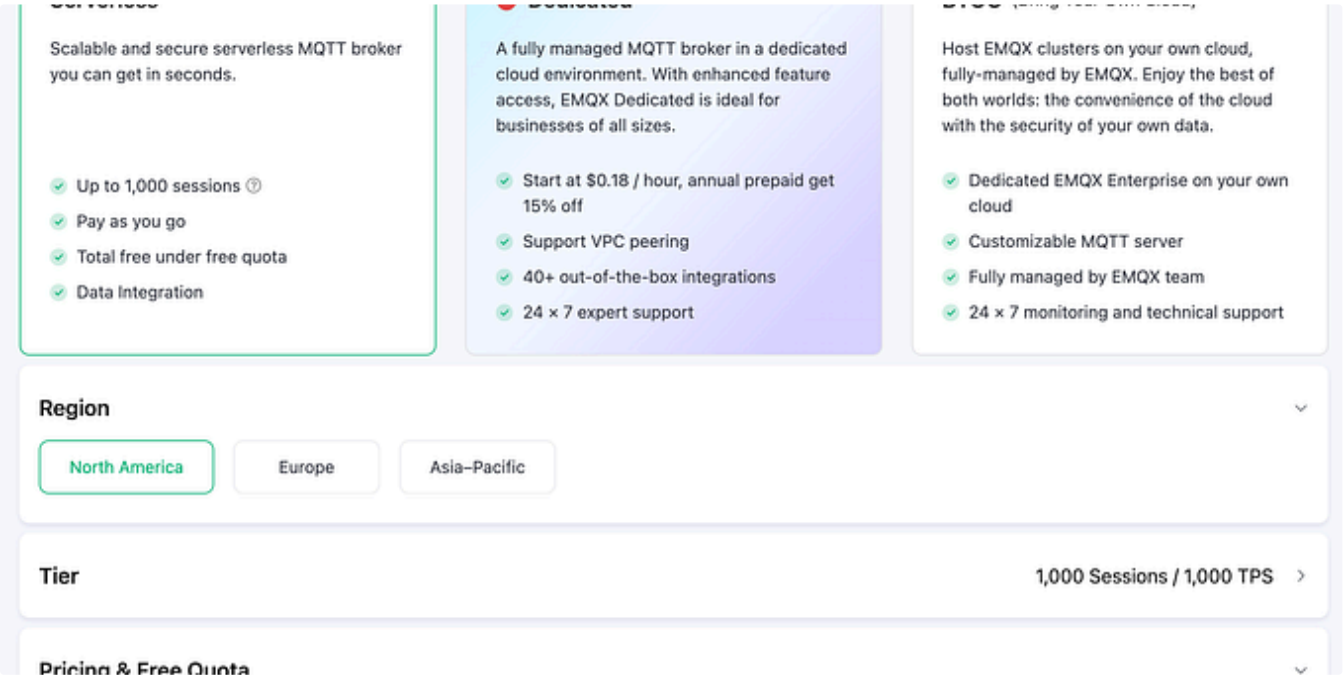
14

Ardian Danny

[OSCP Practice Series 1] Proving Grounds — Exfiltrated

Machine Type: Linux

3 min read · Dec 26, 2023



EMQ Technologies

MQTT with openHAB: A Step-by-Step Tutorial

Introduction

6 min read · Apr 19, 2024

66

See more recommendations