## Weird Python

**Author:** Honeynet and Omoremi Adeleke

**Date:** September 9 2024

**Version:** Forensic Challenge 14

**Type of Dataset:** pcapng

**Software:** VirtualBox, Wireshark, network miner, uncompile2

**Ref:** <https://www.honeynet.org/challenges/forensic-challenge-14-weird-python/>

<https://github.com/honeynet/forensic_challenges/blob/master/README.md>

**Expected results**

* Identify attack method
* Pinpoint infected systems
* List exfiltrated files
* Analyze malware actions
* Classify attack type
* Decompile Python bytecode main.pyc
* Understand main.pyc purpose
* Analyze payload protection
* Understand Pete's departure
* Identify security risks

**Goal of lab**

* Train participants in malware analysis, network traffic analysis, digital forensics, and security awareness
* Provide experience in identifying, analyzing, and mitigating cyber threats
* Identify how the attack was carried out, including any use of DNS spoofing, malware payloads, and race conditions.
* Follow the web traffic of a targeted user (John) to locate infected systems, understand the sequence of events leading to infection, and identify files downloaded during the attack.
* Determine the type and intent of the attack by analyzing both malicious and benign executable files.
* Decompile the Python bytecode (main.pyc) to understand its purpose and assess its functionality.

## CASE

SCENARIO

Your boss John went to a BYOD conference lately. Yeah, he’s that kind of security guy… After some mumble about targeted attacks happening during the event, your team finally got their hands on a PCAP with his traffic. Your colleague Pete Galloway investigated the incident. Yesterday, he casually mentioned that he found some weird Python bytecode but couldn’t make much sense out of “random” payloads yet.

EVIDENCE

* .pcapng file
* Web Pages Accessed
* Requested Resources
* IP Addresses and Hosts
* DNS Responses
* File Downloads
* Transaction ID
* Malicious Python bytecode

## TASK 0 – Environment Setup

OVERVIEW

Each task progressively built upon the findings of the previous one, culminating in a comprehensive understanding of the attack’s source, methods, and implications. The analysis demonstrated the critical importance of network traffic analysis in identifying vulnerabilities and understanding the dynamics of cyber threats.

Start Virtual machine

Getting .pcapng file

* C/P [link](https://www.honeynet.org/challenges/forensic-challenge-14-weird-python/)

<https://www.honeynet.org/challenges/forensic-challenge-14-weird-python/>

* Download Files ‘conference.pcapng (mirror)’
  + The password to unlock the fc14.zip:

e84f2e4cad4d83ba2af9ee8f17412e97bf29787a8ec4675fc1e4717bd3799426

Getting Network Miner (after install, find the .exe in the opt folder)

* Open terminal and follow along with this [link](https://www.netresec.com/?page=Blog&month=2014-02&post=HowTo-install-NetworkMiner-in-Ubuntu-Fedora-and-Arch-Linux) to install Network Miner

<https://www.netresec.com/?page=Blog&month=2014-02&post=HowTo-install-NetworkMiner-in-Ubuntu-Fedora-and-Arch-Linux>

## TASK 1 – Determine the Source of the Attack

OVERVIEW  
The goal of this task is to analyze network traffic captured in the conference.pcapng file to identify the source of an attack and trace the actions of a user named John. The objective involves filtering HTTP GET requests to examine the specific web pages John accessed, including his search for "byod" on Reddit and the links he reviewed. Key takeaways include identifying the role of specific IP addresses in the attack, understanding the implications of DNS spoofing, and recognizing the sequence of requests that led to malicious activity. Ultimately, the task emphasizes the importance of network analysis in uncovering vulnerabilities and understanding the dynamics of cyber threats.

Actions:

* Analyzed the conference.pcapng file using Wireshark.
* Filtered HTTP GET requests to identify specific web pages accessed by John.
* Examined the sequence of requests and the content being requested.

Goals:

* Identify the specific websites John accessed and the search terms he used.
* Trace the sequence of his actions leading to potential malicious activity

**SUBTASK 1.1**

Upload conference.pcapng to wireshark

Apply filter to narrow down GET requests based on the type of content being requested

http.request.method == "GET" && http.accept contains "text/html"

SS

|  |
| --- |
|  |

Select the column Time for filter in ascending order

Request URI (Uniform Resource Identifier) or path specifies the resource or endpoint on the server that the client is requesting

GET /HTTP/1.1

Represents the user having searched for a webpage directly

Using the bottom left window and drop-down Hypertext Transfer Protocol dropdown, have options for viewing the next request and previous request to show incremented steps of the user

Each packet will show the specific path or resource requested by the client to review the http.request.uri fields showing incremented steps of the user

Select each GET /HTTP/1.1, looking in the Hypertext Transfer Protocol dropdown

What 3 webpages did John search for?

|  |
| --- |
|  |

GET /search/

typically indicates that the client is requesting the search functionality or search results from the server

In the list of packets window, find the GET /search/

What did John search for and on what webpage?

|  |
| --- |
|  |

**SUBTASK 1.2**

Select the GET /search/ and while making sure it is select, clear the filter

In the Hypertext Transfer Protocol dropdown, double click on the Next Request in Frame and take note each Full Request URI

Of the byod search results, what 2 links did John review? c/p the description

|  |
| --- |
|  |

Reapply the filter

http.request.method == "GET" && http.accept contains "text/html"

GET /r/

represents the path to the resource on the server that the client wants to access. In this case, /r/ is the specific path being requested

Select the second GET /r/ select Hypertext Transfer Protocol dropdown from the bottom pane

What is the packet number, Host address and the Referrer address? Right click, C/P as description

|  |
| --- |
|  |

Select the second GET /HTTP/1.1 packet(2657)

What does it mean for it to not have a Next request in frame? What is the Host address and the Referrer address?

|  |
| --- |
|  |

**SUBTASK 1.3**

Finally, of all requests from John, there is a download

Explore packet 5675

In the Hypertext Transfer Protocol dropdown the GET is a .zip extension which suggest John downloaded a file

What IP is associated and what is the Host address and the Referrer address?

|  |
| --- |
|  |

Following the original instance of the thewayoftheninga request

Exploring packet 5033

What is the IP address of the first request instance related to thewayoftheninga ?

|  |
| --- |
|  |

Right click on packet 5033 > Follow > http stream

Search the html (in blue) for something possibly related to the last request to download

SS where there might be a possible relation (hint: type ‘Download’ into find)

|  |
| --- |
|  |

Close the follow window

What is the new filter? Why did the filter change

|  |
| --- |
|  |

Begin to confirm if these 2 IPs are related

Apply the filter:

http contains "harveycartel.org"

SS

|  |
| --- |
|  |

To confirm the file type is

Right click the last packet > Follow > TCP Stream

SS

|  |
| --- |
|  |

## TASK 2 – DNS Spoofing

OVERVIEW

This task focuses on analyzing DNS traffic related to the user John to identify vulnerabilities in network security. While HTTP facilitates web content transfer, DNS translates domain names into IP addresses, enabling server connections. We aimed to uncover evidence of DNS spoofing and race conditions by isolating and examining DNS responses. This analysis underscores the importance of robust DNS security measures in defending against cyber threats.

**Actions:**

* Analyzed DNS traffic related to John's activities.
* Applied filters to isolate DNS responses and identified suspicious packets.
* Checked for race conditions and the legitimacy of DNS responses.

**Goals:**

* Understand the role of DNS in the attack and identify any spoofed responses.
* Determine the relationship between legitimate and spoofed IP addresses

**SUBTASK 2.1**

Analyzing the DNS responses

HTTP is about transferring web content, while DNS is about translating domain names into IP addresses so that your browser can find and connect to the correct server

Apply Modified filter

(http.request.method == "GET" && http.accept contains "text/html") or (udp.port == 53 or dns)

Immediately it shows a good amount of additional traffic.

Be sure to keep time ascending and scroll to the bottom.

It shows that there are three DNS packets with the exact same source and destination

In the section scroll the window to the right to see the rest of the info for each packet

SS

NOTE: use this ss to help answer Q2 from the overall honeynet challenge

|  |
| --- |
|  |

Scroll back to the left

Select the packet 10935

What is the url of this response? Hint: (Shown in the bottom pane under the **Queries** drop-down)

Using what IP?

|  |
| --- |
|  |

**SUBTASK 2.2**

In the menu bar select > Analyze > Expert information

2 packets are shown with warning labels and the first packet in the expert info shows an ip address that does not correspond to the ninja game site at IP 54.. nor the ninja game download site IP 81

Right click on the first warning > apply filter > selected

Close

What is the new filter?

What is the unknown IP address? Hint: (look for the **Answers** drop down)

|  |
| --- |
|  |

SS

|  |
| --- |
|  |

This is proof of Trojanized DNS spoof which is achieved by winning a ‘race condition’

What is the time difference by looking at the time codes for the two responses?

What is the transaction id? Hint: (right after the drop-down **Domain Name System (response)**)

Explain the race condition of this case and do include the IP addresses to identify systems?

What IP was retransmitted and received first?

How can it be that the original website responded first but the spoofed ip was able to hijack?

|  |
| --- |
|  |

Conclude this task Identifying the IP addresses and urls and their roles collected (3)

|  |  |  |
| --- | --- | --- |
| IP Address | URL | Role (original or spoofed) |
|  |  |  |
|  |  |  |
|  |  |  |

## TASK 3 – Analyze Network Traffic After Execution

OVERVIEW

The objective is to identify extracted files, user-agent strings, and potential unauthorized data exfiltration through HTTP POST requests. Key findings include recognizing the specifics of successful and failed file uploads, along with understanding the underlying attack method, such as DNS spoofing. Additionally, the task prompts reflection on the nature of the attack, determining whether it was targeted or automated. The overall takeaway emphasizes the importance of network analysis in detecting malicious activity and enhancing cybersecurity awareness

Actions:

* Used NetworkMiner to analyze network traffic after John executed a file download.
* Identified User-Agent strings and extracted files from HTTP POST requests.
* Reviewed successful and failed file upload attempts.

Goals:

* Recognize any unauthorized data exfiltration through HTTP POST requests.
* Analyze the specifics of file uploads to understand potential data loss.

SUBTASK 3.1

Use Wireshark to save .pcapng file as a .pcap file

Go to File > Export Specified Packets (or File > Save As if you want to save all packets).

In the "Save as type" dropdown, select pcap - Wireshark/tcpdump/

Choose a location to save the .pcap file, enter a filename **‘confr’** for example, and click Save.

The file is in the .pcap format, which is compatible with Network Miner Free version

Run NetworkMiner

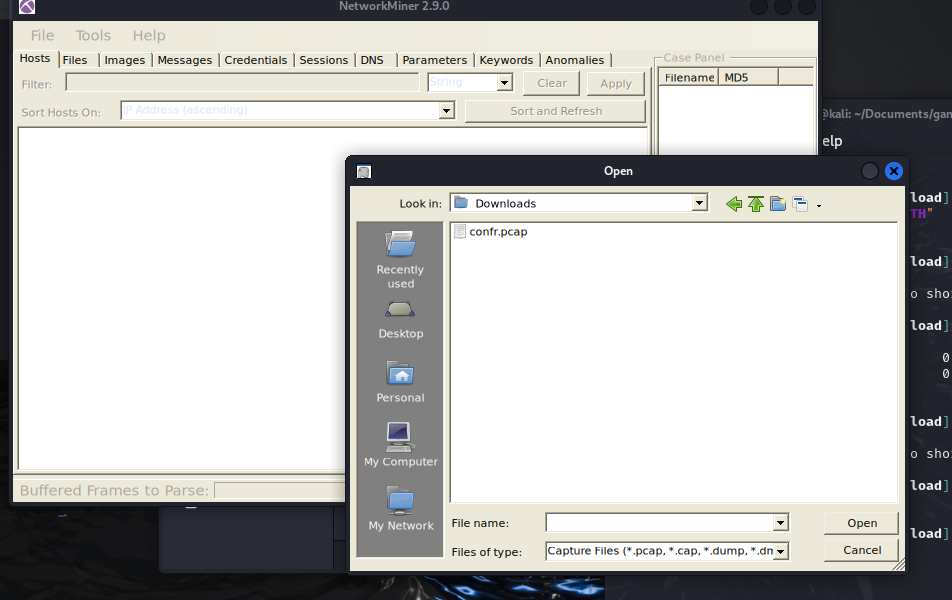
mono NetworkMiner.exe --noupdatecheck

Or

Run .exe from the opt folder

File > Open > select the confr.pcap file

**Preview ->**



Select the Files tab, scroll to the bottom

It shows a bunch of ‘submit\_highscores....’ files immediately preceding the download of the Nv2-PC.exe file

Scroll the window to the see the details section and a bunch of HTTP POST details show being sent from the Users own IP address of 127... to the attackers IP address of 81...

SUBTASK 3.2

Apply filter: submit

SS

|  |
| --- |
|  |

A User-Agent can help identify specific types of traffic, such as those originating from browsers or certain devices, which can be useful in detecting unauthorized file exfiltration or malicious activity

Use the Parameters tab to inspect HTTP requests and find User-Agent strings to identify the files extracted in each of the POST sessions

To determine the User-Agent

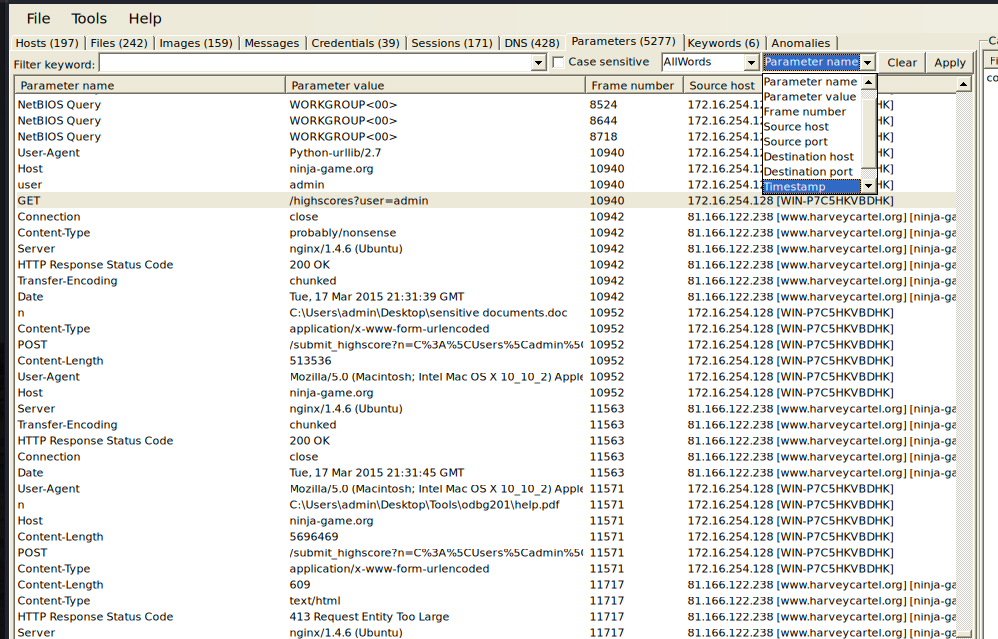
Select the Parameters tab – again scroll all the way to bottom

From here it displays not only the submit\_... files but also more info about each HTTP POST

What is the User-Agent of the HTTP POST instances IN relation to the submit\_highscore files?

Right click > C/P Parameter Value

|  |
| --- |
|  |



To determine the Files Extracted

NetworkMiner shows files by their actual name if the file name is included in HTTP requests or responses

The Parameter Name ‘n’ is being used as a parameter to reference a file path/name in the HTTP request to upload/POST

What are the four files extracted? Which Upload failed and why?

|  |
| --- |
|  |

## TASK 4 – Analyze The Executable Downloaded

OVERVIEW

Python bytecode presence in a PCAP might indicate malicious activity. Attackers sometimes deliver malicious Python scripts over the network in the form of bytecode for various exploits such as remote code execution or malware using Python.

Actions:

* Analyzed the downloaded executable files, focusing on both benign and malicious content.
* Extracted and decompiled the contents of the executable files using appropriate tools.
* Reviewed strings from the malicious executable to ascertain its functionality.

Goals:

* Determine the nature of the downloaded files and identify any malicious components.
* Analyze the executable to understand its functionality and potential impact.

SUBTASK 4.1

In network miner > files > filter: exe > apply > right click > open folder

Take a note of all the different files within this download

Extract > select Documents > create new folder called Task3

/opt/NetworkMiner\_2-9/AssembledFiles/81.166.122.238/TCP-80/nv2/

Right click > start.cmd > open with > vim or notepad

Paste contents or SS

|  |
| --- |
|  |

Explain what this represents

|  |
| --- |
| n\_v14.exe is ... |

OPEN terminal

$ pip install uncompyle6

CONFIRM installation

$ uncompyle6 –help

ADD to path if command not found

$ export PATH=$PATH:/home/kali/.local/bin

$ echo 'export PATH=$PATH:~/.local/bin' >> ~/.zshrc

$ uncompyle6 --help

uncompyle6: command not found

$ source ~/.zshrc

$ uncompyle6 --help

Usage: uncompyle6 [OPTIONS] FILES...

SUBTASK 4.2

EXTRACTING all files in Nv2-PC.exe

$ cd /opt/NetworkMiner\_2-9/AssembledFiles/81.166.122.238/TCP-80/nv2

$ opt/…/AssembledFiles/81.166.122.238/TCP-80/nv2]

$ unrar x Nv2-PC.exe

$ ls

The ls should now show all files in the Nv2-PC.exe

EXTRACTING readable strings from MAIN.exe (Q6 is it normal)

$ strings main.exe > /home/kali/Documents/TaskE3/outMainX.txt

OPEN this file

The strings command is a utility in Unix-like operating systems (including Linux and macOS) as well as Windows that extracts printable strings from binary files, such as executables, libraries, or other non-text files

List 3 strings that were extracted that related to Python

|  |
| --- |
|  |

The malware seems to be written in Python. Is this “normal” Python?

|  |
| --- |
| YES  main.exe is ... |

SUBTASK 4.3

DECOMPYLING main.pyc

$ uncompyle6 -o /home/kali/Documents/TaskE3/ main.pyc

Should show:  
 main.pyc --

Successfully decompiled filed

Python compiles scripts before running them and stores the compiled version in pyc files

At the simple level, a .pyc file is a binary file containing only three things:

A four-byte magic number,

A four-byte modification timestamp, and

A marshalled code object

From File System, go to Documents TaskE3

Right click open main.py

To answer Honeynet Q7: What does main.pyc do?

SS & explain

|  |
| --- |
| SS:  This script is meant to .. |

## TASK 5 – Lab Report on Network Traffic Analysis and Cyber Attack Investigation

Writing a lab report is crucial as it documents experiments and ensures findings can be replicated. It communicates research methods and results to peers, fostering collaboration and knowledge sharing. The process promotes critical analysis of data, deepening understanding of the subject matter, while also evaluating students' grasp of scientific concepts. Furthermore, it helps develop skills in technical writing and data analysis, making lab reports essential for advancing scientific knowledge and preparing individuals for future research or professional roles.

The report-writing process should be completed in the following order:

* + - Analyze the data and review the examiner notes
    - Write the first draft
    - Revise the draft and create the final report

Write a lab report and fill in these sections:

Abstract (summary)

Table of contents

Body of report

Conclusion

|  |
| --- |
| **Abstract**    **Table of Contents**  **1. Introduction**  **2. Task 1: Determine the Source of the Attack**  **3. Task 2: Analyzing the DNS Responses**  **4. Task 3: Analyze Network Traffic After Execution**  **5. Task 4: Analyze the Executable Downloaded**  **6. Conclusion**    **Body of Report**    1. Introduction    2. Task 1: Determine the Source of the Attack  **Overview**    **Findings**    3. Task 2: Analyzing the DNS Responses  **Overview**    **Findings**    4. Task 3: Analyze Network Traffic After Execution  **Overview**  **User-Agent:**  Mozilla/5.0 (Macintosh; Intel Mac OS X 10\_10\_2) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/40.0.2214.115 Safari/537.36    **Extracted Files**    **Failed Upload**    5. Task 4: Analyze the Executable Downloaded  **Overview**    **Extracted Files**    **Decompiling**:    **Conclusion** |

## Q&A

Answering The Honeynet Challenge Questions

1. BYOD seems to be a very interesting topic. What did your boss do during the conference?

|  |
| --- |
| Search reddit for byod and look up games and download one |

1. What method did the attacker use to infect your boss? Which systems (i.e. IP addresses) are involved?

|  |
| --- |
| DNS spoofing - The race condition in this case is that the malicious attacker spoofed the harveycartel.org original IP 207.150.212.43 to instead be the attackers IP for 81.166.122.238 so they share the same transaction id as a response to john request for game download IP 207.150.212.43 was first. This response can be hijacked if the user's device might not have recognized this response due to the presence of the subsequent spoofed response ie. The Transactions ID  IP Addresses Involved:  54.231.10.92  81.166.122.238  207.150.212.43 |

1. Based on the PCAP, which files were exfiltrated? List the filenames.

|  |
| --- |
| sensitive documents.doc  EmiratesETicket1.pdf  EmiratesETicket2.pdf |

1. Can you sketch an overview of the general actions performed by the malware?

|  |
| --- |
|  |

1. Do you think this is a targeted or an automated attack? Why?

|  |
| --- |
| It is likely the specific target was the BYOD conference   * no referrer header for the request to thewayoftheninja.org, indicating the game may have been sent via email from a bookmark, non-tracking browser advertised at the conference, or word of mouth * The attacker relied on timing their fake DNS responses suggesting that they only had access to a device on the local network because the attack could not stop the DNS request from reaching 8.8.8.8 |

1. The malware seems to be written in Python. Is this “normal” Python? What’s different?

|  |
| --- |
| YES  main.exe is clearly written in python originally and when completed, was made into an EXE |

1. What does main.pyc do?

|  |
| --- |
| This script is meant to detect when the .pyc is being ran under a debugger and if detected, prints the message on line 7 |