# PROJECT REPORT ON

# ADVANCED WIRESHARK NETWORK FORENSICS

Submitted in partial fulfilment of the requirements for the award of the degree of

## BACHELOR OF TECHNOLOGY

**Submitted by**

**122004028 - AVINASH R – B.TECH ECE**


**School of Computing**
**SASTRA DEEMED TO BE UNIVERSITY**
(A University established under section 3 of the UGC Act, 1956)
Tirumalaisamudram
Thanjavur - 613401
December-2020

**SHANMUGHA**
**ARTS, SCIENCE, TECHNOLOGY & RESEARCH ACADEMY (SASTRA DEEMED TO BE UNIVERSITY)**
**(A University Established under section 3 of the UGC Act, 1956)**
**TIRUMALAISAMUDRAM, THANJAVUR – 613401**


## BONAFIDE CERTIFICATE

Certified that this project work entitled **"ADVANCED WIRESHARK NETWORK FORENSICS"** submitted to the Shanmugha Arts, Science, Technology & Research Academy (SASTRA Deemed to be University), Tirumalaisamudram - 613401 by AVINASH R (REG NO. 122004028),B.TECH ECE in partial fulfillment of the requirements for the award of the degree of **BACHELOR OF TECHNOLOGY** in their respective program. This work is an original and independent work carried out under my guidance, during the period September2020 - December 2020.

Submitted for Project-based Viva Voce held on **16.02.2021**

**Examiner – I** **Examiner – II**

# TABLE OF FIGURES

# ABBREVIATIONS

**IP----------→ Internet protocol**

**TCP ----------→ Transmission control protocol**

**FTP----------→ File transfer protocol**

**HTTP----------→ Hyper Text transfer protocol**

**ARP----------→Address resolution protocol**

**HXD----------→ Hex editor**

**PCAP----------→ Packet capture**

**DOS----------→ Disc-based operating system**

**SYN-ACK----------→ Synchronize acknowledge**

**DNS----------→ Domain name system**

**IOC----------→ Indicators of compromise**

# ABSTRACT

In our rapidly evolving world, Computer Networks has become a very pre-dominant and intricate field of technology. At the same time, hackers across the globe are designing and inflicting various attacks through the internet for different reasons like machine corruption, hijacking & Information theft.
Wireshark is the most widely used network and protocol analyzer, while it is also a very essential tool for Network Forensics.
Network administrators should be able to investigate and analyze the network traffic to understand what is happening and to deploy immediate response in case of an identified attack.
Wireshark can be used in identifying and categorizing various types of attack signatures.
The purpose of this project is to demonstrate how Wireshark can be applied for network protocol diagnosis.
We will see how Wireshark is used to perform Network Forensics and solve real-world problems.

In this project, we will go through the basic terminologies and jargons that we would need to be familiar with for Network Forensic analysis. Then,
We will cover our first real-world problem, where we will look at network traffic and try to understand a few things about the virus, next we will reassemble the network bites to pull out the virus and analyze it.
Then, we are going to build on what we learnt and cover another real-world scenario where we are going to look at a series of attacks on both network level and application level and learn how to weed out the signal from the noise.

The reason for picking this topic is that there is not a lot of content on packet analysis & advanced network forensics, even the ones available do not cover the deep intricate details of it.
This project tries to build an understanding of the process as a whole than mere definitions and base level procedures.

# TABLE OF CONTENTS

# ACKNOWLEDGEMENTS

First of all, I would like to thank God Almighty for his endless blessings.

I would like to express my sincere gratitude to Dr S. Vaidyasubramaniam, Vice-Chancellor for his encouragement during the span of my academic life at SASTRA Deemed University.

I would forever remain grateful and I would like to thank Dr Thenmozhi K, Dean, School of Electrical and Electronics Engineering and R. Chandramouli, Registrar for their overwhelming support provided during my course span in SASTRA Deemed University.

I would specially thank and express my gratitude to Prof.Sasikala Devi .N, Senior Assistant Professor, School of Computing and Prof.Parvathy A, Assistant professor for providing me an opportunity to do this project and for her guidance and support to successfully complete the project.

I also thank all the Teaching and Non-teaching faculty, and all other people who have directly or indirectly helped me through their support, encouragement and all other assistance extended for completion of my project and for successful completion of all courses during my academic life at SASTRA Deemed University.

Finally, I thank my parents and all others who helped me acquire this interest in the project and aided me in completing it within the deadline without much struggle.

# INTRODUCTION

Ever wondered how to track an attacker through a network?

Wireshark is a great tool for network analysis, it is prominent for analysis and troubleshooting of network traffic and there are plenty of rudimentary uses of Wireshark.
But instead, Wireshark can be used to perform Network Forensics analysis and solve real-world problems.
Understanding of protocol analyzers can help us bridge the gap between basic theory and real - world problems.
Forensics by definition has a specific meaning, it is to collect and prepare evidence for the purpose of litigation, while
Network Forensics is the capture, recording, and analysis of network events in order to discover the source of security attacks or other problem incidents.
We are going to analyze data collected by packet capture which is a part of network forensics as a whole.
Other areas are log analysis and net-flow data which aren't required for our analysis.
In the first place, we are going to get familiar with the definitions and processes that are going to be needed for analysis and investigation.
Basic understanding of below mentioned software and protocols is must:

1. Protocol Analyzers (Wireshark, TCP dump)
2. OSI and TCP/IP model
3. Major protocols (HTTP, DNS, TCP, UDP etc.)

   Packet capture forensics analysis is downright depended on knowing 'what to look for?' and 'where to look?' to properly analyze a file within a reasonable amount of time.

   Must have tool are:

   1. Wireshark
   2. Hex Editor ( Hxd is more preferred for windows)

Optional tool:

1. Network miner
2. Scalpel
3. GeoIP database

Both these tool are used to understand how to manually do network based file carving, both these tolls will help speed up the process and make it a little easier to deal with larger pcap files.
Also add a few columns to Wireshark to make it easier to point out things that are needed.

# The Network Forensics Process

1. Know your triggering events
2. Have a goal
3. Packet capture analysis (draw conclusions and identify indicators of compromise)

Typically when presented with a capture file for analysis, so before getting started we have to know what has happened i.e what is the triggering event, after finding the triggering event, a series of focus goals have to designed, these goals will help focus the investigation into achievable results
Finally, begin with the packet capture analysis.
At the end of this entire process, we'd have identified things looking for and allow us to draw conclusions and take appropriate actions

# Triggering Events

When trying to understand the triggering events, these things might prompt an investigation:

1. IDS alert
2. Log or network anomalies (i.e DoS or virus activity)
3. Known malicious/compromised systems
4. Time frame, etc.

# Have a Goal

1. Always have a goal for analysis ( not having a goal could prolong a particular investigation)
2. Prioritize the goal

Looking at network packet captures is like looking for many needles in many haystacks, that is why knowing at least that what we are looking at is something important than just random information is very necessary.

# PCAP Analysis Methodology

1. Pattern matching

   This is where we have to identify and filter packets of interest by matching specific values or protocol metadata

2. List conversations

   From there we can list all the conversation streams within the filtered captured data

3. Isolate and export

   Then we can isolate and export specific conversation streams of interest.

4. Draw conclusions

   Once isolated we can extract files or data streams to compile the data and draw conclusions based on it.

This is an iterative process which means that we may be going back to previous steps often as we find new data.

That is it, we are now ready to go forth and analyze the capture files.
Next we are going to see how all of this comes together in the first Real-World scenario.

# PROPOSED FRAMEWORK

## SCENARIO 1:

A system is infested with malware and the antivirus on the system did not attack it and the malware has managed to lock up the system.

## TRIGGERING EVENTS:

1. User reporting malware activity
2. Current AV solution does not have a signature for the virus; nor is the virus recoverable from the infected host.

### What we know:

1. Full network packet capture for the day of the incident.
2. Host of interest : 12.183.1.55

## GOALS:

### IOC: indicators of compromise

1. Where did the user contract the malware from?
2. Malware file( if possible)
3. See the malwares activity on the system
   (What kind of calls to the internet does it make?)
4. Does it try to self-propagate through the internal network?
   (Is it a worm or a virus?)
5. Possible network traffic signatures that we can use to catch other systems potentially infected with the same piece of malware?

# Wireshark setup:

Open the pcap file in Wireshark and the first thing to do is to add a few columns.

Right click on any column →column preferences →click on the plus sign and add two columns

Details for the first column are:

Title: stream ID

Fields: tcp.stream

Details for the second stream are:

Title: host

Fields: http.host

Place ‘stream ID’ between protocol and length and place ‘host’ between length and info.

Write the goals in a notepad.

Now that we have our Wireshark setup and goals written, network analysis can be started.

See **fig.1** for reference for the above process

# Pattern matching:

We already know the IP address of the system we are interested in, so let us create a display filter to show us only the traffic related to that device

Type in:

“ ip.addr == 12.183.1.55 ”

Notice anything that looks suspicious.

An important pointer when investigating capture files is:

“What may be true for one network may not be true for another network”

In our pcap file, a ‘.ru’ domain name might seem suspicious but it is also possible that it is a multi-national company.

Either way let us check it,

Right click on it →follow → TCP stream

Few things we note from it:

1. There is no user agent.

   It is kind-of strange because when normally using a web-browser to make a web request, the browser includes its user agents in the web headers.
2. It is an exe file, we can also find by seeing the first few bytes of the file signature.

Now write down the found out info in the notepad.

See **Fig.2** for reference.

Now,

To find the file type for the file signature noted.

Go to google→ file signature database → garykessler.net

See **Fig.3** for reference.

Now that we have the bytes of the file, how to pull it with Wireshark:

1. Change the traffic we are looking at to only include the communication coming from the server to the client.
2. Show the raw bytes instead of the ASCII form.
3. Save the file as dump1.
   (not as .exe because it is a live virus)

See **Fig. 4** for reference.

To get the original file, we need to strip off any and all protocol headers and footers.

In this case, to deal with the headers, we need Hex Editor.

Open the file in Hex Editor.

The way HTTP headers work is they let us when headers end and the data starts by

"0D 0A 0D 0A "

So just delete everything before that.

See **Fig.5** for reference.

When done, the file will start will MZ, the first few bytes of the file signature.

Now save the file.

Before saving the file, we have to switch off the antivirus in the system because otherwise the antivirus would recognize and put the file in quarantine.

Immediately after carving out information in this investigation, we have to get the hash of the file.
For which, we will use hash my file and get MD5 and SHA-256,doesn't matter whichever but make sure to get a value so that the process is repeatable so that the carved file and the original file can be matched together.
See **Fig.6** for reference.

# File Analysis

Now for file analysis, we have to use VirusTotal.

" **VirusTotal** is an online service that analyzes files and URLs enabling the detection of viruses, worms, Trojans and other kinds of malicious content using antivirus engines and website scanners. It also can be used to detect false positives. "

Upload the file on VirusTotal and we will get details about the virus.
There is also a list of antivirus vendors where they note which AV can and cannot detect the virus.
See **Fig.7** for reference.

So we were able to collect and analyze the malware file, next we have to look at what kind of network traffic it generates.
We can filter out the virus download by filtering out TCP stream 5.
Write " ip.addr == 12.183.1.55 && !(tcp.stream ==5) "
We get a lot of DNS traffic, which seems to be somewhat randomly generated domain names.
See **Fig.8** for reference.

When scrolling down, quite a few 'syn' packets can be seen being sent, meaning they were all sent out in a short amount of time.
We would start to see connections that were being established and then closed immediately, this is pretty standard form for botnet persistence.
The virus comes with a preloaded list of domains or with a built-in way to generate domain names,
It then reaches out to each domain names in the list of find out what is available online.
That way if some of the domains in the list are blocked or shut down it still has a way to come home

Let us analyze a packet to find what TCP is trying to communicate on:

To see which of these domains it is connected to and it stayed connected to, very likely that it would be using http web traffic, so let us check the host headers.

Now write

"ip.addr == 12.183.1.55 && !(tcp.stream ==5) && http.host

We can see that most of the communication happens with one website.

See **Fig.9** for reference.

Now,

Right click on host→follow→TCP stream

To find out what kind of traffic is present.

It looks like a normal webpage, might have redirected the user to their site to buy a fake antivirus

Note all the information in the notepad.

See **Fig.10** for reference.

Last thing is if the virus tries to propagate over the network like a worm.

If the virus tries to reach out to other devices on the network, it might try to follow RFC 1918 and look for private IP addresses, there is also a chance that it takes the IP addresses of the infected machine and tries to reach out to other devices on the network as well.

RFC 1918 is the standard for private IP addressing.

We are going to build one large filter

The filter design is,

"ip.src == 12.183.1.55 &&(ip.addr == 192.168.0.0/16 || ip.addr == 172.16.0.0/12 || ip.addr == 10.0.0.0/8 || ip.dst == 12.0.0.0/8)) "

Store it in Wireshark, and we might get ICMP destination unreachable messages,

See **Fig.11** for reference.

This is actually a bug in the Wireshark filters where it thinks that the ICMP messages are sourced and destined for the same address.

Add a no ICMP to the end of the filter.

“&&!(icmp)”

At this point it does not look like there are any attempts from the virus to try and connect to other internal systems.
See **Fig.12** for reference.

Write this information in notepad too.

# Let us now review what we have found.

1. Where did the user contract the malware from?

   User made a direct call to the executable. There-fore, user either deliberately downloaded the malware, or there was a piece of malware sleeping on the system.

2. Malware file:

   Malware file has been carved out and analyzed via virustotal.com

   Write the MD5 and SHA256 hash values.

3. What kind of calls to the internet does it make?

   -DNS queries for a number of domains

   -HTTP communication for websites located on a few of those domains.

4. Does it try to self-propagate?

   No communication to other internal addresses.

- Network traffic signatures:

  High volume of DNS queries within a short amount of time.

# SCENARIO 2:

Client Network Attack

## TRIGGERING EVENTS:

1. A denial of service (Dos) attack has been reported against FTP server 192.168.56.1
2. FTP traffic spikes were seen prior to the FTP server being taken offline

## WHAT WE KNOW:

1. Captured traffic data this is narrowed down between an attacking host (192.168.56.101) and the FTP server (192.168.56.1).

## WHAT WE WANT TO FIND OUT:

A little more abstract;

1. What caused the spike in FTP traffic?
2. What events took place prior to the FTP server being taken offline?

   (Example, Were any files transferred to/from the FTO server or were any user accounts compromised)

Before getting started, document the goals, steps and results.

In this scenario, the goals are a little bit abstract and will depend on what we find in the 'pcap'.

We want to know:

-What led up to the attack?

-What types of attacks did the attacker perform?

-Were they able to get in and what did they find?

Now that we have goals, open the 'pcap' file in Wireshark.
A lot of ARP frames are being sent, in most of these ARP requests the IP address is going up for each request.
First, take a look at how many conversations are within the capture file to begin with to find how many IP addresses are in this.
Go to;
Statistics→conversations
We can see that it has been already filtered for us.

Glancing at the TCP tab, notice that over 7000 TCP connections were made just between the two addresses.
See **Fig.13** for reference.

→We know that this capture file has only two IP addresses, even then let us look through the attack traffic and try to find out what the attacker was able to see on our network.
Look at the ARP scans again, ARP scans work by sending out a bunch of ARP requests throughout the network, when another device on the network receives an ARP request, it will send an ARP reply.
Let us filter it down to show only the ARP replies.

We notice that there is a third address '.100', we don't have any information on it, it is possible that there was no communication from the attacker at all (or)

It could have been filtered out from the capture.

Make a note of the data found in the notepad.
See **Fig.14** for reference.

We are done with the ARP traffic, so let us filter it out.
Now we are starting to see a flood of packets coming from our attacker and based on the changing port numbers, it is pretty obviously a **network port scan.**

To figure out what open ports the attacker was able to find in their scan of the system

Go to:

Transmission control protocol → flags → apply as filter → selected

When we send out a SYN, we expect to see a SYN ACK returned,

So let us filter by packets with the SYN ACK flag set and we are able to see the ports the attacker was able to find open.

Document all the data found.

See **Fig.15** for reference.

Still have a lot more data to come through.

We know this is a FTP server, so eliminate the obvious and filter out port 21;

With that we can separate the signal from the noise and verify that these are the only open ports the attacker was able to find.

"Following a formal methodology doesn't just mean we will get the answer we want, also means we will get the context of those answers "

Now look at the FTP traffic at port 21,

We have already seen the traffic that's part of stream-20, it was the port scan we have documented earlier.

But after, we see a lot of connection requests going straight to port 21.

Follow one of the streams and analyze,

Looks like a bunch of login attempts, hit the up arrow few times and check out the other streams

It definitely looks like a brute force attack.

See **Fig.16** for reference.

To figure if they did get in,

We see the FTP codes of incorrect Log- in, check out in google for the correct login FTP code.

It is FTP response code 230.

Put that in the filter and analyze,

We get only streams and both of them have the response 230.

Follow TCP streams for both the streams.

In the first one, they logged in but didn't go anywhere with it.
In the second one,

- They logged in with 'anon/anon'
- Listed the directories
- Changed to images
- Listed directory again
- Downloaded file "why we can't have nice cat.png "

Document the details found in the notepad.
See **Fig.17** for reference.

Since this is all over the network,

We can see the results of each command, hit the up arrow couple of times and step through the streams.

We see the contents of the first, second directory and then PNG that the attacker downloaded.

(Match with the first few bytes of file signature to find out the type of file).

See **Fig.18** for reference.

"In real-world capture files, it won't always be as easy as pressing the up arrow and streams list, sometimes the next streams or next several streams are actually part of other traffic, this is why we have to filter packets and list conversations as the first step when analyzing pcap files"

Another way to find the downloaded file is to look in the conversation lists, we know the size of the file given by FTP server response, so find a conversation that has at least the same number of bytes.

Then, filter by that and look at the stream.

See **Fig.19** for reference.

Save the '.png' file in raw form.
See **Fig.20** for reference.

Now that we have located the file, carve it out like in scenario 1 and take a hash of it.

That way, compare the hash and the hash of the original file on the server for integrity.

Document the hash data found.
See **Fig.21** for reference.

Last step,
Go ahead and open the file to see what the attacker was able to get his hands on.
See **Fig.22** for reference.

# Investigation results

Attacker first initiated a ARP scan of the subnet
192.168.56.0/24

→ Following hosts were discovered:
192.168.56.1 &
192.168.56.100

Attacker then began a port scan of host 192.168.56.1

➔ The following ports were found open:
21, 445, 139, 135, 49152, 49153, 49154, 49155, 49156.

Attacker followed up with an FTP brute force attack against FTP servers.

➔ The credentials **anon/anon** were compromised.

Attacker successfully logged in as user anon with stolen credentials.

➔ File “whycantwehaveanicecat.png” was downloaded.

➔ MD5 sum of the file has been found.

# SCREENSHOTS

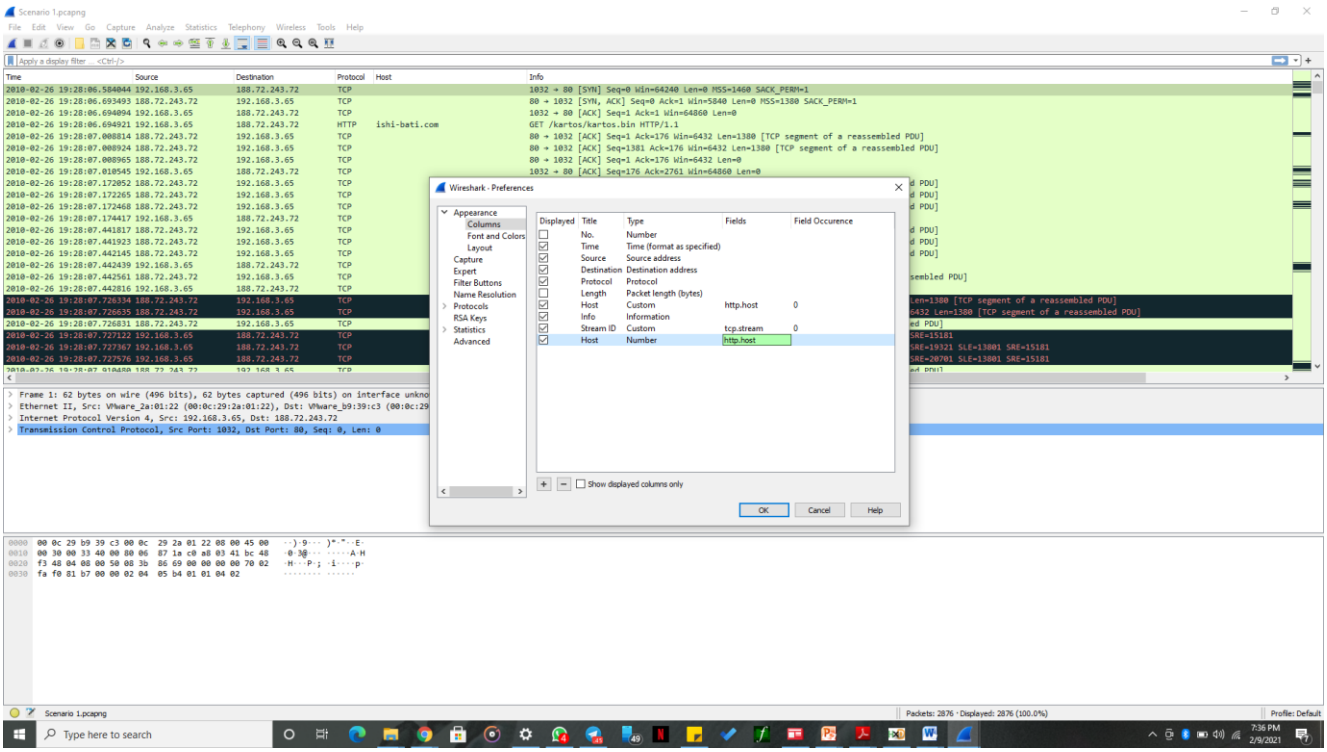


**Fig.1**

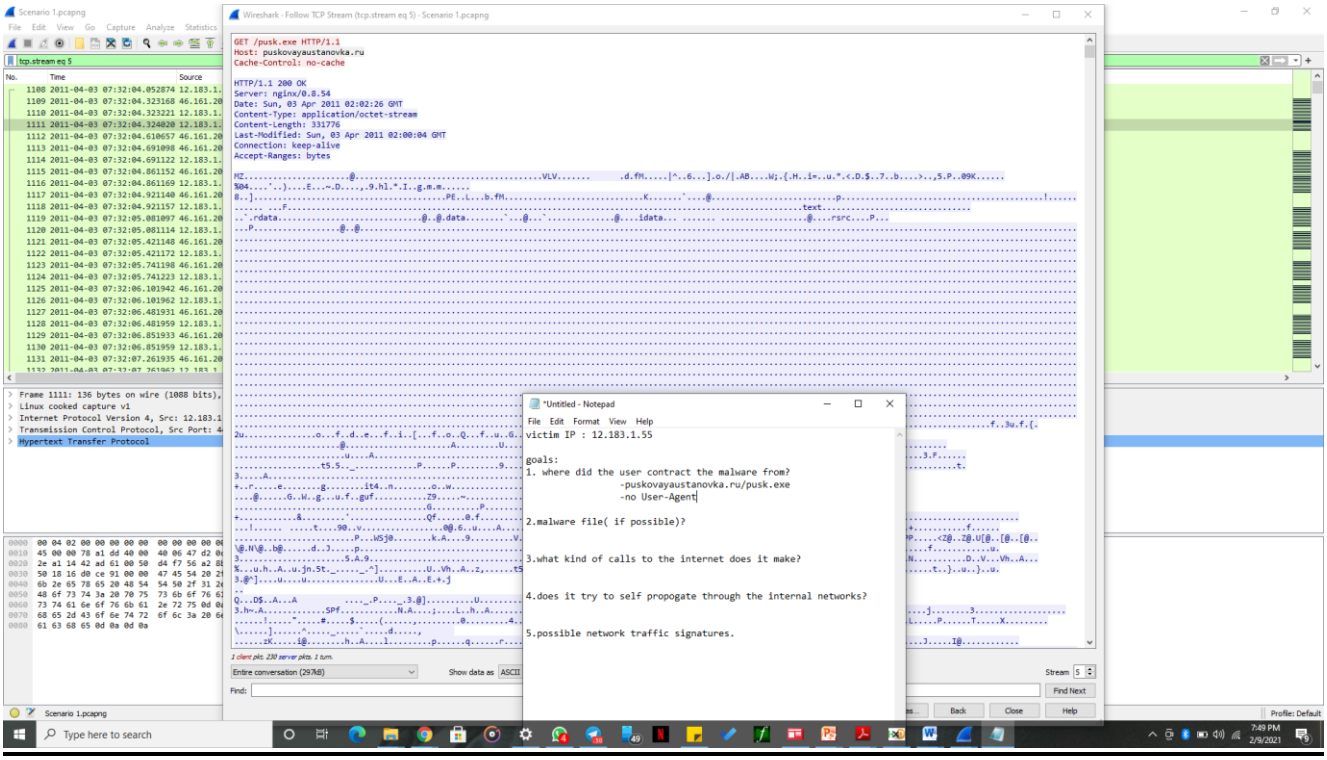


**Fig.2**

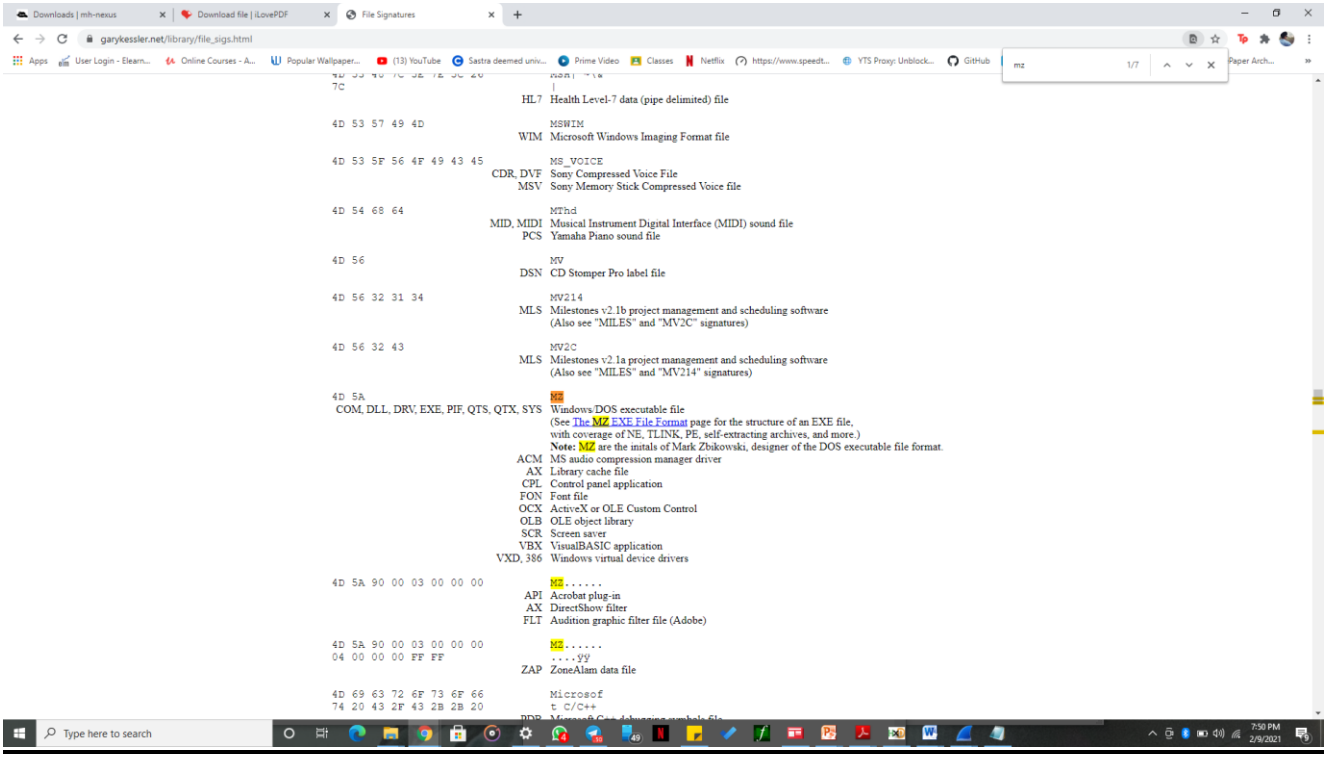
7C
HL7 Health Level-7 data (pipe delimited) file
4D 53 57 49 4D
MSWIM
WIM Microsoft Windows Imaging Format file
4D 53 5F 56 4F 49 43 45
MS_VOICE
CDR, DVF Sony Compressed Voice File
MSV Sony Memory Stick Compressed Voice file
4D 54 68 64
MThd
MID, MIDI Musical Instrument Digital Interface (MIDI) sound file
PCS Yamaha Piano sound file
4D 56
MV
DSN CD Stomper Pro label file
4D 56 32 31 34
MV214
MLS Milestones v2.1b project management and scheduling software
(Also see "MILES" and "MV2C" signatures)
4D 56 32 43
MV2C
MLS Milestones v2.1a project management and scheduling software
(Also see "MILES" and "MV214" signatures)
4D 5A
MZ
COM, DLL, DRV, EXE, PIF, QTS, QTX, SYS Windows/DOS executable file
(See The MZ EXE File Format page for the structure of an EXE file,
with coverage of NE, TLINK, PE, self-extracting archives, and more.)
Note: MZ are the initals of Mark Zbikowski, designer of the DOS executable file format.
ACM MS audio compression manager driver
AX Library cache file
CPL Control panel application
FON Font file
OCX ActiveX or OLE Custom Control
OLB OLE object library
SCR Screen saver
VBX VisualBASIC application
VXD, 386 Windows virtual device drivers
4D 5A 90 00 03 00 00 00
MZ.....
API Acrobat plug-in
AX DirectShow filter
FLT Audition graphic filter file (Adobe)
4D 5A 90 00 03 00 00 00
04 00 00 00 FF FF
MZ.....
....yy
ZAP ZoneAlarm data file
4D 69 63 72 6F 73 6F 66
74 20 43 2F 43 2B 2B 20
Microsof
t C/C++
Type here to search

**Fig.3**

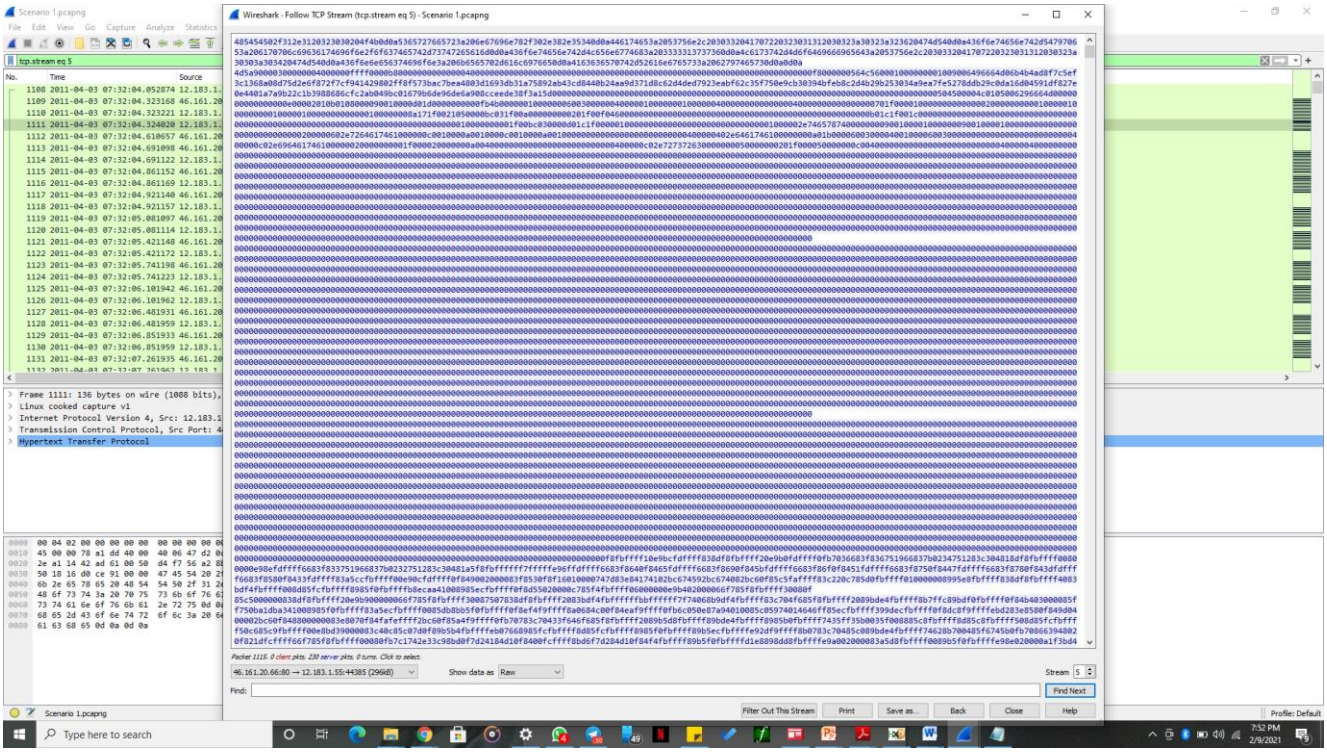
Wireshark · Follow TCP Stream (tcp.stream eq 5) · Scenario 1.pcapng
Type here to search

**Fig.4**

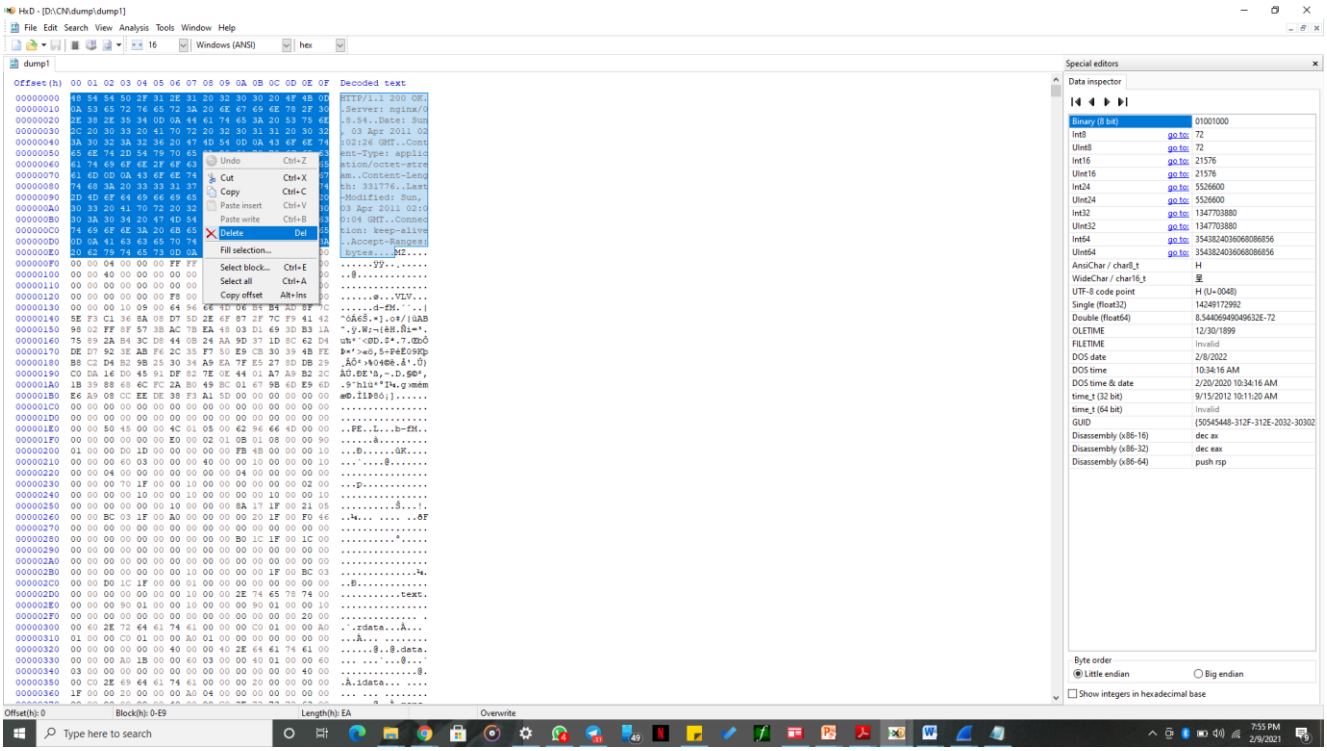
HxD - [D:\CN\dump\dump1]
File Edit Search View Analysis Tools Window Help
dump1
Offset (h) 00 01 02 03 04 05 06 07 08 09 0A 0B 0C 0D 0E 0F Decoded text
Undo Ctrl+Z
Cut Ctrl+X
Copy Ctrl+C
Paste insert Ctrl+V
Paste write Ctrl+B
Delete Del
Fill selection...
Select block... Ctrl+E
Select all Ctrl+A
Copy offset Alt+Ins
Special editors
Data inspector
Byte order
Little endian
Big endian
Show integers in hexadecimal base
Type here to search
7:55 PM
2/9/2021

**Fig.5**

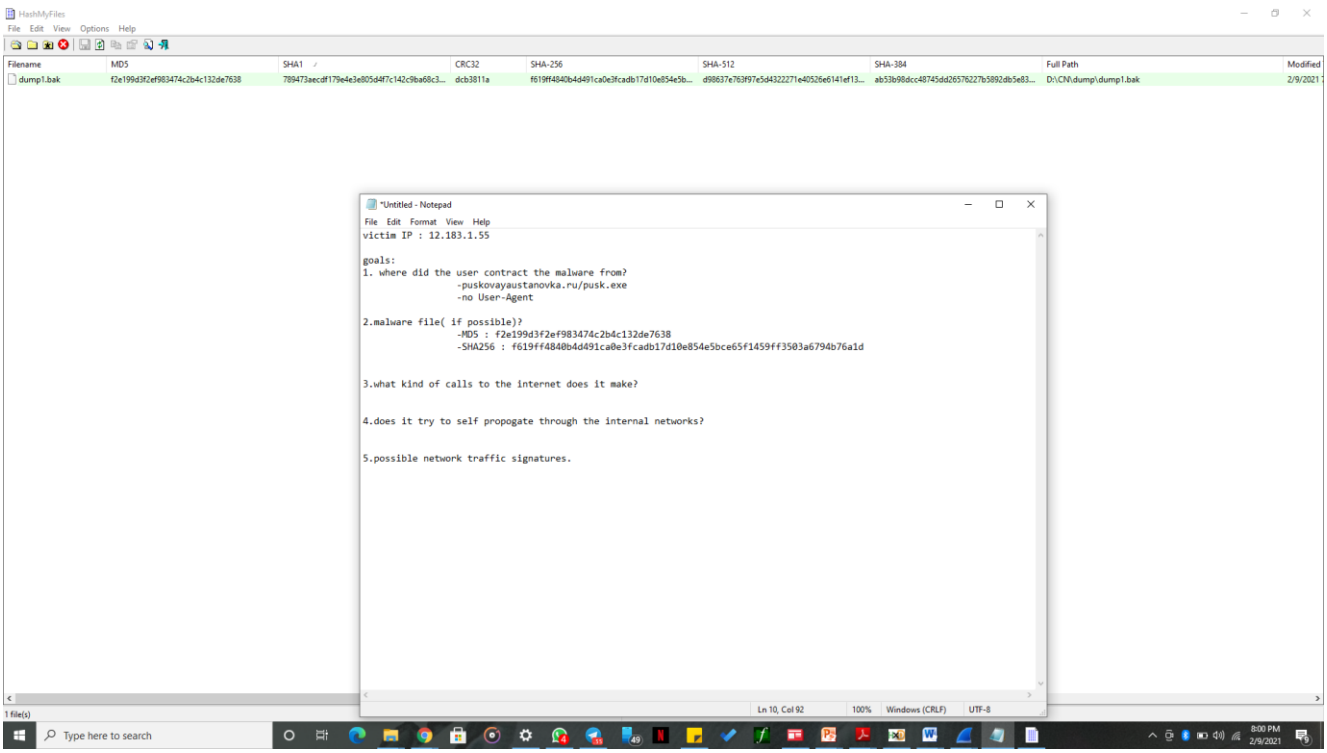
HashMyFiles
File Edit View Options Help
Filename MD5 SHA1 CRC32 SHA-256 SHA-512 SHA-384 Full Path Modified
dump1.bak f2e199d3f2ef983474c2b4c132de7638 789473aecdf179e4e3e805d4f7c142c9ba68c3... dcb3811a f619ff4840b4d491ca0e3fcadb17d10e854e5b... d98637e763f97e5d4322271e40526e6141ef13... ab53b98dcc48745dd26576227b5892db5e83... D:\CN\dump\dump1.bak 2/9/2021
"Untitled - Notepad
File Edit Format View Help
victim IP : 12.183.1.55
goals:
1. where did the user contract the malware from?
-puskovayaustanovka.ru/pusk.exe
-no User-Agent
2.malware file( if possible)?
-MD5 : f2e199d3f2ef983474c2b4c132de7638
-SHA256 : f619ff4840b4d491ca0e3fcadb17d10e854e5bce65f1459ff3503a6794b76a1d
3.what kind of calls to the internet does it make?
4.does it try to self propogate through the internal networks?
5.possible network traffic signatures.
Ln 10, Col 92
100%
Windows (CRLF)
UTF-8
1 file(s)
Type here to search
8:00 PM
2/9/2021

**Fig.6**

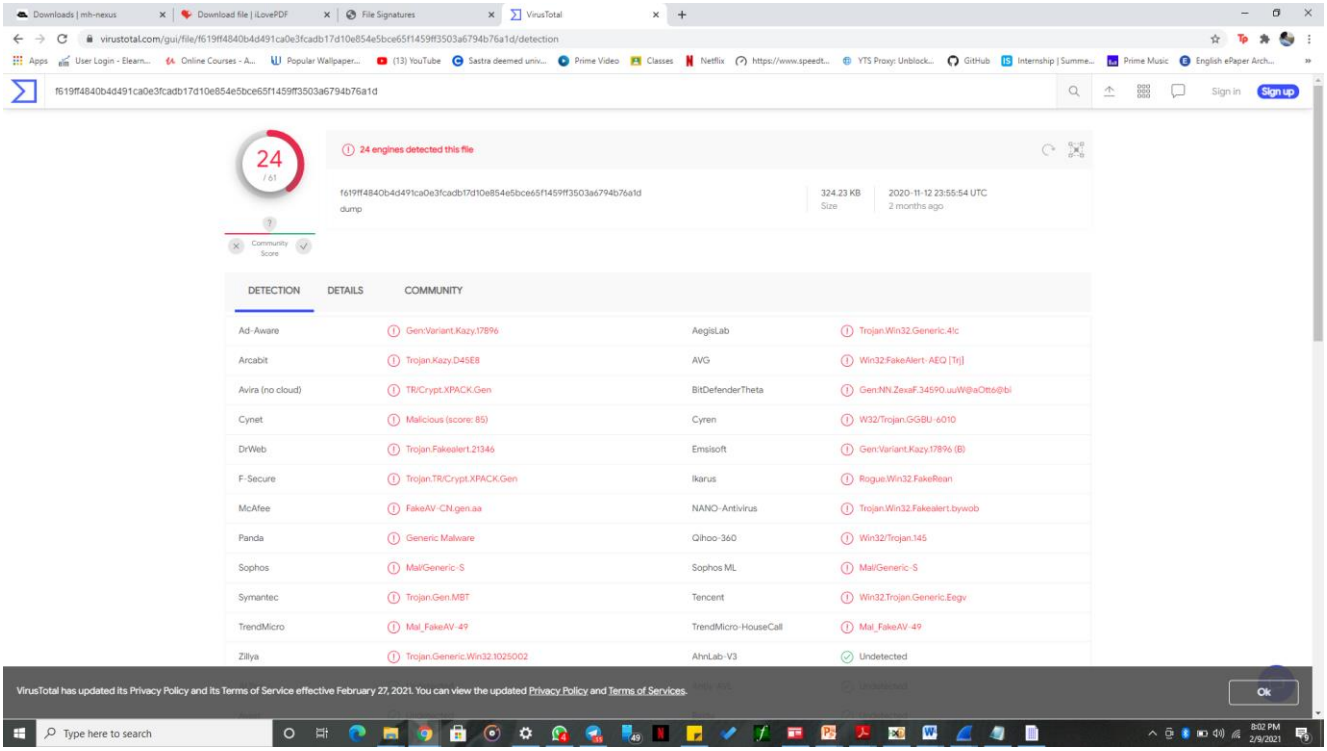
VirusTotal
f619ff4840b4d491ca0e3fcadb17d10e854e5bce65f1459ff3503a6794b76a1d
24
/ 61
24 engines detected this file
f619ff4840b4d491ca0e3fcadb17d10e854e5bce65f1459ff3503a6794b76a1d
dump
324.23 KB
Size
2020-11-12 23:55:54 UTC
2 months ago
DETECTION
DETAILS
COMMUNITY
Ad-Aware
Gen:Variant.Kazy.17896
AegisLab
Trojan.Win32.Generic.4!c
Arcabit
Trojan.Kazy.D45E8
AVG
Win32:FakeAlert-AEQ [Trj]
Avira (no cloud)
TR/Crypt.XPACK.Gen
BitDefenderTheta
Gen:NN.ZexaF.34590.uuW@aOtt6@bi
Cynet
Malicious (score: 85)
Cyren
W32/Trojan.GGBU-6010
DrWeb
Trojan.Fakealert.21346
Emsisoft
Gen:Variant.Kazy.17896 (B)
F-Secure
Trojan.TR/Crypt.XPACK.Gen
Ikarus
Rogue.Win32.FakeRean
McAfee
FakeAV-CN.gen.aa
NANO-Antivirus
Trojan.Win32.Fakealert.bywob
Panda
Generic Malware
Qihoo-360
Win32/Trojan.145
Sophos
Mal/Generic-S
Sophos ML
Mal/Generic-S
Symantec
Trojan.Gen.MBT
Tencent
Win32.Trojan.Generic.Eegv
TrendMicro
Mal_FakeAV-49
TrendMicro-HouseCall
Mal_FakeAV-49
Zillya
Trojan.Generic.Win32.1025002
AhnLab-V3
Undetected
VirusTotal has updated its Privacy Policy and its Terms of Service effective February 27, 2021. You can view the updated Privacy Policy and Terms of Services.
Ok

**Fig.7**

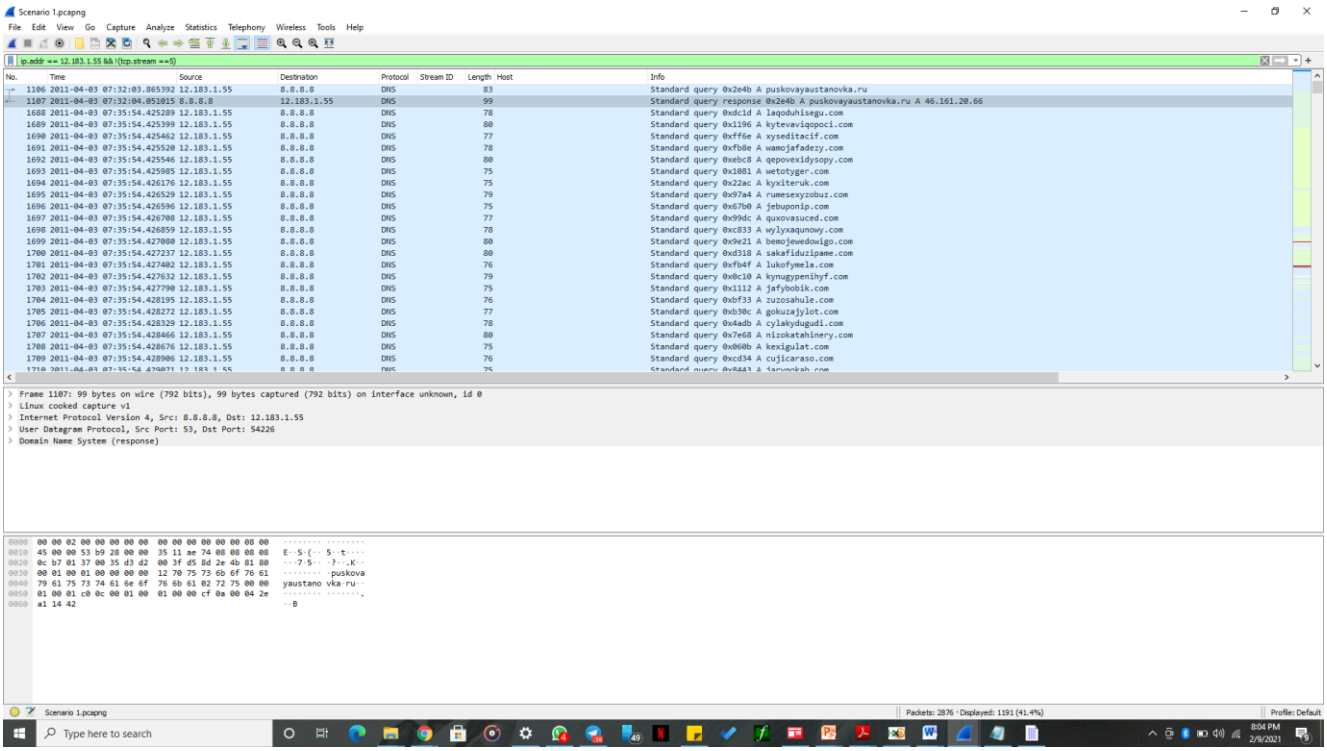
Scenario 1.pcapng
File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help
ip.addr == 12.183.1.55 && !(tcp.stream ==5)
No. Time Source Destination Protocol Stream ID Length Host Info
1106 2011-04-03 07:32:03.865392 12.183.1.55 8.8.8.8 DNS 83 Standard query 0x2e4b A puskovayaustanovka.ru
1107 2011-04-03 07:32:04.051015 8.8.8.8 12.183.1.55 DNS 99 Standard query response 0x2e4b A puskovayaustanovka.ru A 46.161.20.66
1688 2011-04-03 07:35:54.425289 12.183.1.55 8.8.8.8 DNS 78 Standard query 0xdc1d A laqoduhisegu.com
1689 2011-04-03 07:35:54.425399 12.183.1.55 8.8.8.8 DNS 80 Standard query 0x1196 A kytevaviqopoci.com
1690 2011-04-03 07:35:54.425462 12.183.1.55 8.8.8.8 DNS 77 Standard query 0xff6e A xyseditacif.com
1691 2011-04-03 07:35:54.425520 12.183.1.55 8.8.8.8 DNS 78 Standard query 0xfb8e A wamojafadezy.com
1692 2011-04-03 07:35:54.425546 12.183.1.55 8.8.8.8 DNS 80 Standard query 0xebc8 A qepovexidysopy.com
1693 2011-04-03 07:35:54.425985 12.183.1.55 8.8.8.8 DNS 75 Standard query 0x1081 A wetotyger.com
1694 2011-04-03 07:35:54.426176 12.183.1.55 8.8.8.8 DNS 75 Standard query 0x22ac A kyxiteruk.com
1695 2011-04-03 07:35:54.426529 12.183.1.55 8.8.8.8 DNS 79 Standard query 0x97a4 A rumesexyzobuz.com
1696 2011-04-03 07:35:54.426596 12.183.1.55 8.8.8.8 DNS 75 Standard query 0x67b0 A jebuponip.com
1697 2011-04-03 07:35:54.426708 12.183.1.55 8.8.8.8 DNS 77 Standard query 0x99dc A quxovasuced.com
1698 2011-04-03 07:35:54.426859 12.183.1.55 8.8.8.8 DNS 78 Standard query 0xc833 A wylyxaqunowy.com
1699 2011-04-03 07:35:54.427080 12.183.1.55 8.8.8.8 DNS 80 Standard query 0x9e21 A bemojewedowigo.com
1700 2011-04-03 07:35:54.427237 12.183.1.55 8.8.8.8 DNS 80 Standard query 0xd318 A sakafiduzipame.com
1701 2011-04-03 07:35:54.427402 12.183.1.55 8.8.8.8 DNS 76 Standard query 0xfb4f A lukofymela.com
1702 2011-04-03 07:35:54.427632 12.183.1.55 8.8.8.8 DNS 79 Standard query 0x0c10 A kynugypenihyf.com
1703 2011-04-03 07:35:54.427790 12.183.1.55 8.8.8.8 DNS 75 Standard query 0x1112 A jafybobik.com
1704 2011-04-03 07:35:54.428195 12.183.1.55 8.8.8.8 DNS 76 Standard query 0xbf33 A zuzosahule.com
1705 2011-04-03 07:35:54.428272 12.183.1.55 8.8.8.8 DNS 77 Standard query 0xb30c A gokuzajylot.com
1706 2011-04-03 07:35:54.428329 12.183.1.55 8.8.8.8 DNS 78 Standard query 0x4adb A cylakydugudi.com
1707 2011-04-03 07:35:54.428466 12.183.1.55 8.8.8.8 DNS 80 Standard query 0x7e68 A nizokatahinery.com
1708 2011-04-03 07:35:54.428676 12.183.1.55 8.8.8.8 DNS 75 Standard query 0x060b A kexigulat.com
1709 2011-04-03 07:35:54.428906 12.183.1.55 8.8.8.8 DNS 76 Standard query 0xcd34 A cujicaraso.com
Frame 1107: 99 bytes on wire (792 bits), 99 bytes captured (792 bits) on interface unknown, id 0
Linux cooked capture v1
Internet Protocol Version 4, Src: 8.8.8.8, Dst: 12.183.1.55
User Datagram Protocol, Src Port: 53, Dst Port: 54226
Domain Name System (response)
0000 00 00 02 00 00 00 00 00 00 00 00 00 00 00 08 00
0010 45 00 00 53 b9 28 00 00 35 11 ae 74 08 08 08 08
0020 0c b7 01 37 00 35 d3 d2 00 3f d5 8d 2e 4b 81 80
0030 00 01 00 01 00 00 00 00 12 70 75 73 6b 6f 76 61
0040 79 61 75 73 74 61 6e 6f 76 6b 61 02 72 75 00 00
0050 01 00 01 c0 0c 00 01 00 01 00 00 cf 0a 00 04 2e
0060 a1 14 42
Scenario 1.pcapng
Packets: 2876 · Displayed: 1191 (41.4%)
Profile: Default

**Fig.8**

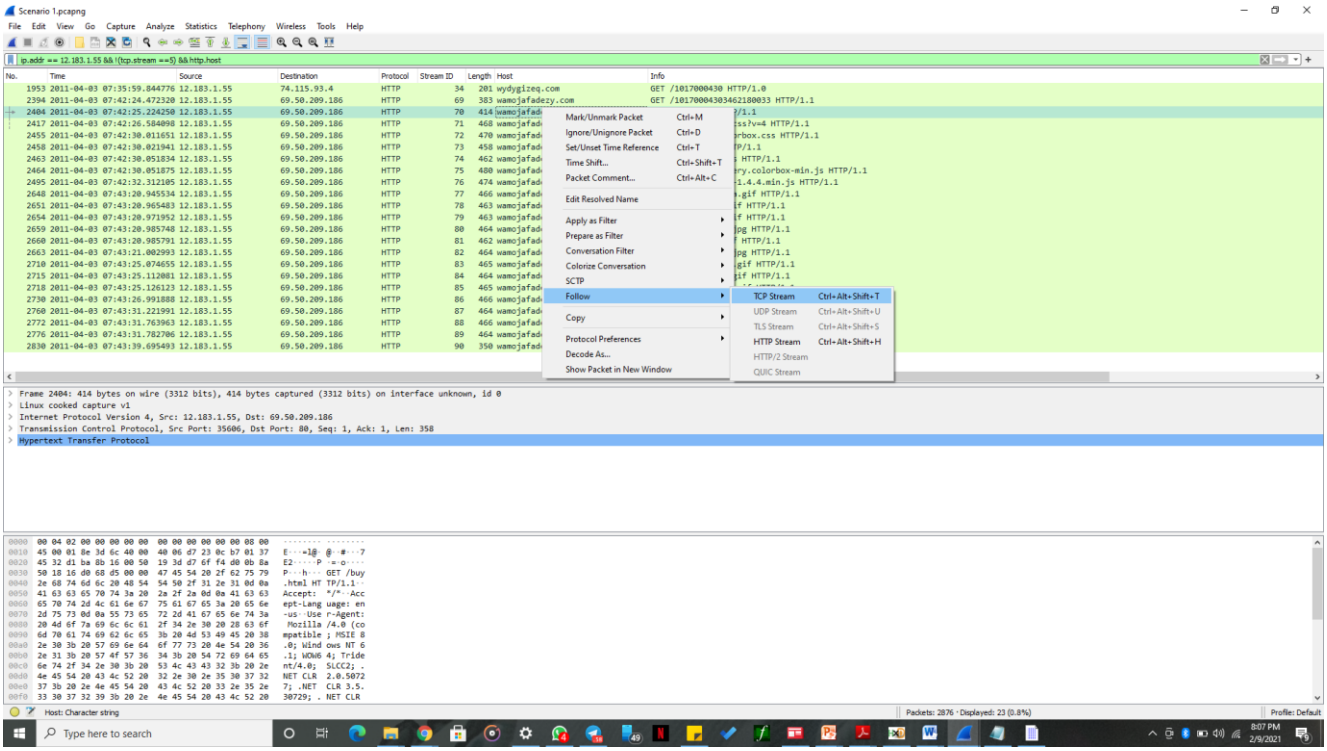

**Fig.9**

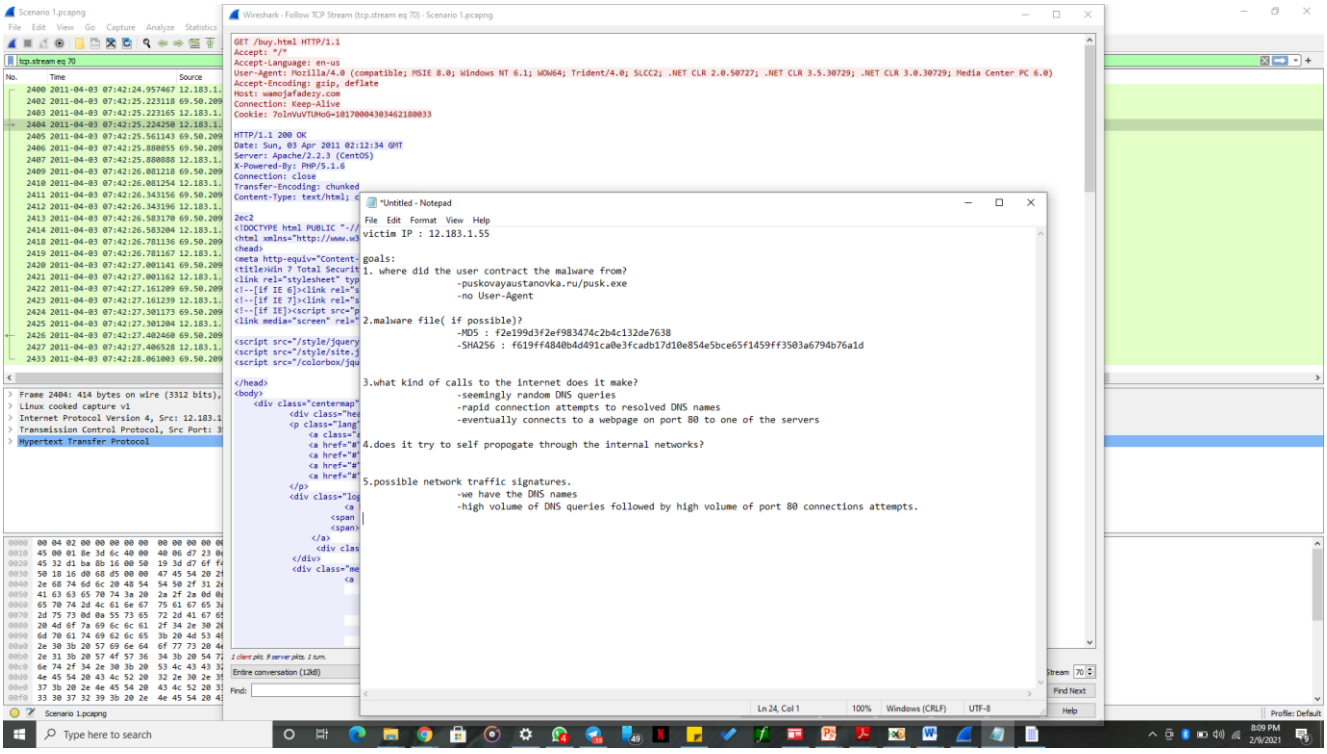

**Fig.10**

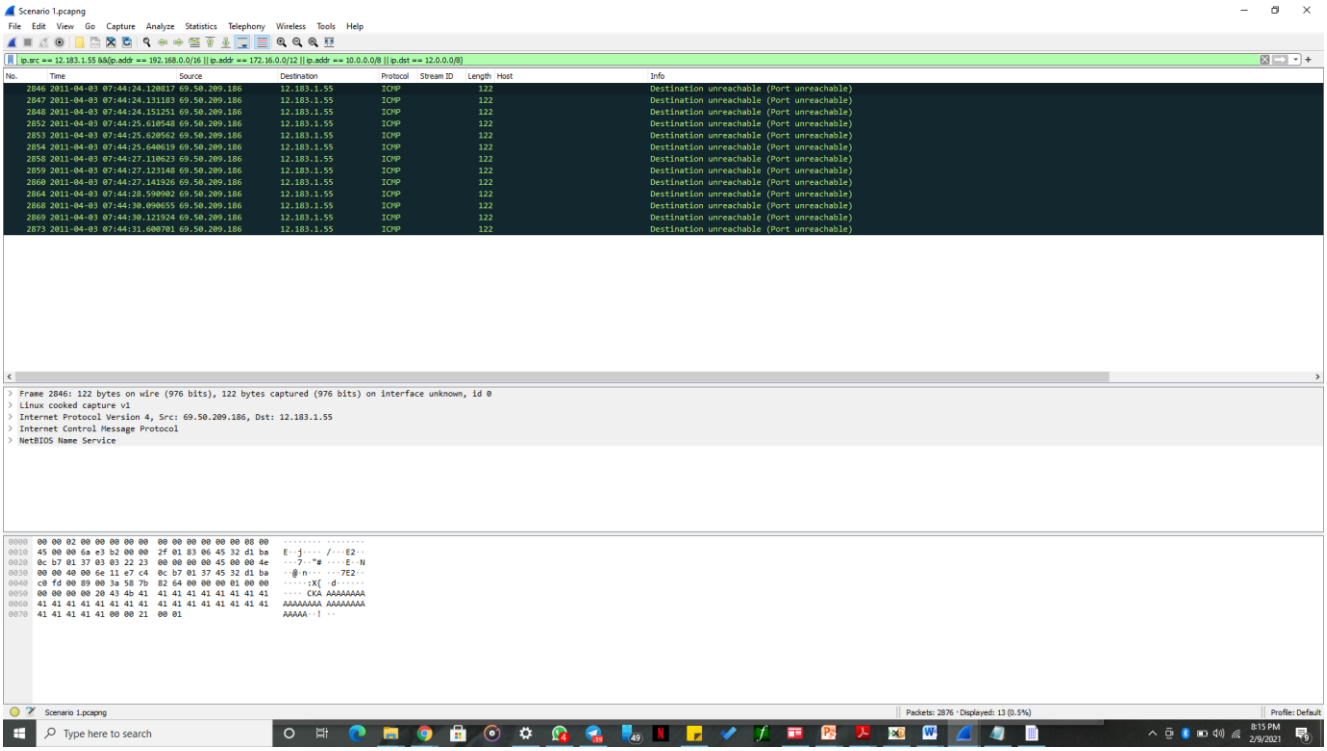
Scenario 1.pcapng
File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help
ip.src == 12.183.1.55 && (ip.addr == 192.168.0.0/16 || ip.addr == 172.16.0.0/12 || ip.addr == 10.0.0.0/8 || ip.dst == 12.0.0.0/8)
No. Time Source Destination Protocol Stream ID Length Host Info
2846 2011-04-03 07:44:24.120817 69.50.209.186 12.183.1.55 ICMP 122 Destination unreachable (Port unreachable)
2847 2011-04-03 07:44:24.131183 69.50.209.186 12.183.1.55 ICMP 122 Destination unreachable (Port unreachable)
2848 2011-04-03 07:44:24.151251 69.50.209.186 12.183.1.55 ICMP 122 Destination unreachable (Port unreachable)
2852 2011-04-03 07:44:25.610548 69.50.209.186 12.183.1.55 ICMP 122 Destination unreachable (Port unreachable)
2853 2011-04-03 07:44:25.620562 69.50.209.186 12.183.1.55 ICMP 122 Destination unreachable (Port unreachable)
2854 2011-04-03 07:44:25.640619 69.50.209.186 12.183.1.55 ICMP 122 Destination unreachable (Port unreachable)
2858 2011-04-03 07:44:27.110623 69.50.209.186 12.183.1.55 ICMP 122 Destination unreachable (Port unreachable)
2859 2011-04-03 07:44:27.123148 69.50.209.186 12.183.1.55 ICMP 122 Destination unreachable (Port unreachable)
2860 2011-04-03 07:44:27.141926 69.50.209.186 12.183.1.55 ICMP 122 Destination unreachable (Port unreachable)
2864 2011-04-03 07:44:28.590902 69.50.209.186 12.183.1.55 ICMP 122 Destination unreachable (Port unreachable)
2868 2011-04-03 07:44:30.090655 69.50.209.186 12.183.1.55 ICMP 122 Destination unreachable (Port unreachable)
2869 2011-04-03 07:44:30.121924 69.50.209.186 12.183.1.55 ICMP 122 Destination unreachable (Port unreachable)
2873 2011-04-03 07:44:31.600701 69.50.209.186 12.183.1.55 ICMP 122 Destination unreachable (Port unreachable)
Frame 2846: 122 bytes on wire (976 bits), 122 bytes captured (976 bits) on interface unknown, id 0
Linux cooked capture v1
Internet Protocol Version 4, Src: 69.50.209.186, Dst: 12.183.1.55
Internet Control Message Protocol
NetBIOS Name Service
0000 00 00 02 00 00 00 00 00 00 00 00 00 00 00 08 00
0010 45 00 00 6a e3 b2 00 00 2f 01 83 06 45 32 d1 ba
0020 0c b7 01 37 03 03 22 23 00 00 00 00 45 00 00 4e
0030 00 00 40 00 6e 11 e7 c4 0c b7 01 37 45 32 d1 ba
0040 c0 fd 00 89 00 3a 58 7b 82 64 00 00 00 01 00 00
0050 00 00 00 00 20 43 4b 41 41 41 41 41 41 41 41 41
0060 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41
0070 41 41 41 41 41 00 00 21 00 01
Scenario 1.pcapng
Packets: 2876 · Displayed: 13 (0.5%)
Profile: Default
Type here to search
8:15 PM
2/9/2021

**Fig.11**

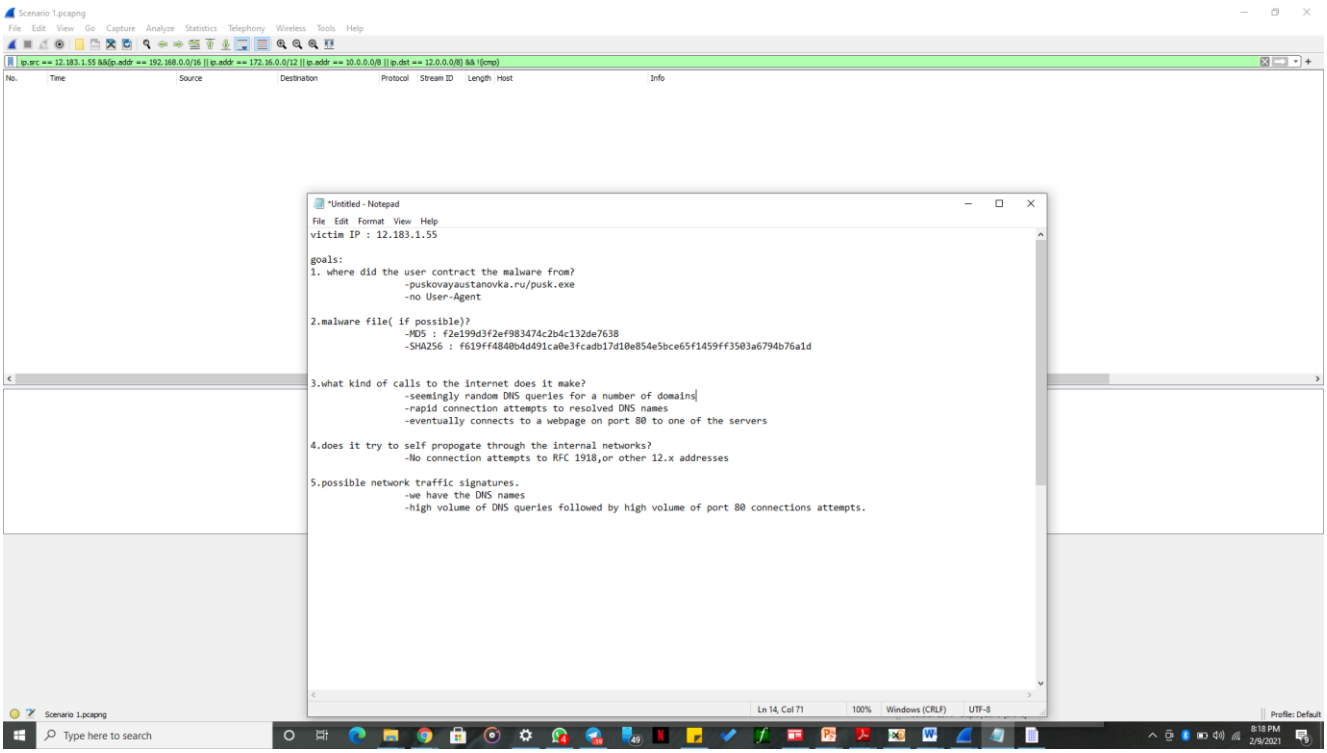
Scenario 1.pcapng
File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help
ip.src == 12.183.1.55 && (ip.addr == 192.168.0.0/16 || ip.addr == 172.16.0.0/12 || ip.addr == 10.0.0.0/8 || ip.dst == 12.0.0.0/8) && !(icmp)
No. Time Source Destination Protocol Stream ID Length Host Info
*Untitled - Notepad
File Edit Format View Help
victim IP : 12.183.1.55
goals:
1. where did the user contract the malware from?
-puskovayaustanovka.ru/pusk.exe
-no User-Agent
2.malware file( if possible)?
-MD5 : f2e199d3f2ef983474c2b4c132de7638
-SHA256 : f619ff4840b4d491ca0e3fcadb17d10e854e5bce65f1459ff3503a6794b76a1d
3.what kind of calls to the internet does it make?
-seemingly random DNS queries for a number of domains
-rapid connection attempts to resolved DNS names
-eventually connects to a webpage on port 80 to one of the servers
4.does it try to self propogate through the internal networks?
-No connection attempts to RFC 1918,or other 12.x addresses
5.possible network traffic signatures.
-we have the DNS names
-high volume of DNS queries followed by high volume of port 80 connections attempts.
Ln 14, Col 71
100%
Windows (CRLF)
UTF-8
Scenario 1.pcapng
Profile: Default
Type here to search
8:18 PM
2/9/2021

**Fig.12**

# SCENARIO-2

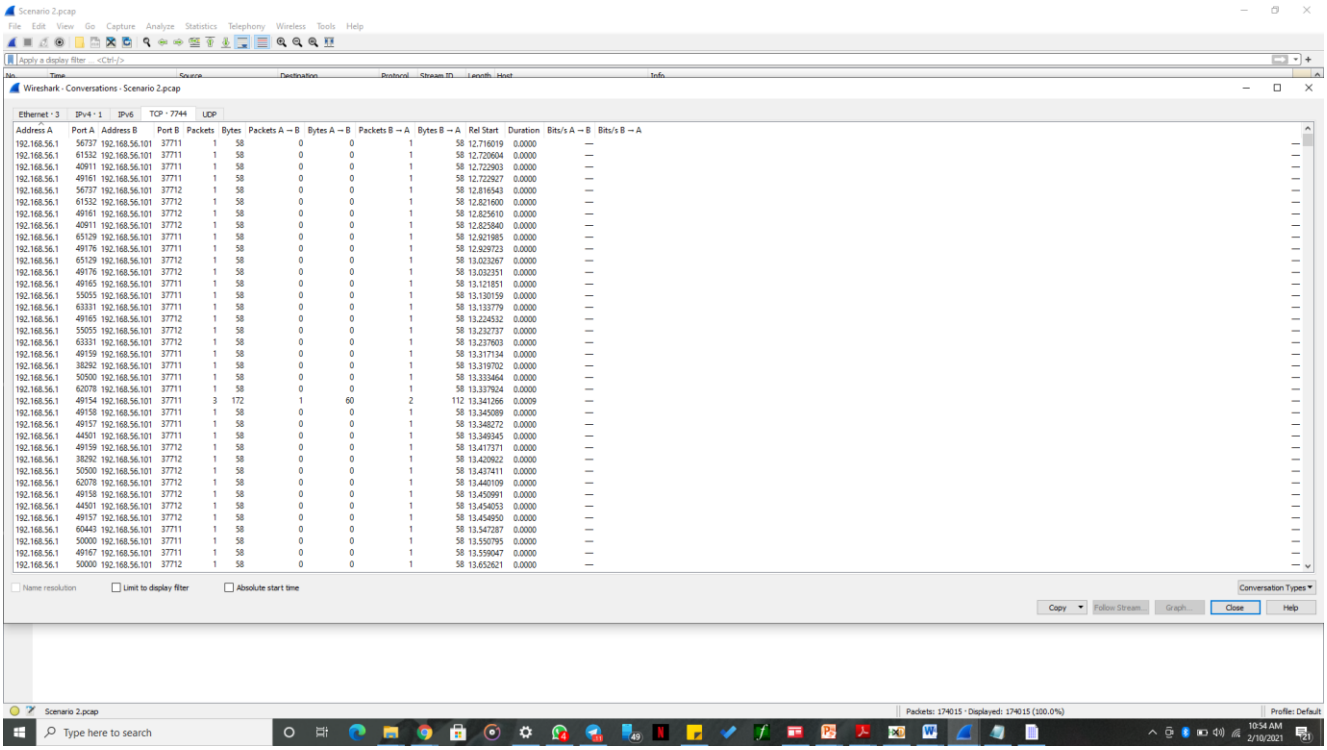

**Fig.13**

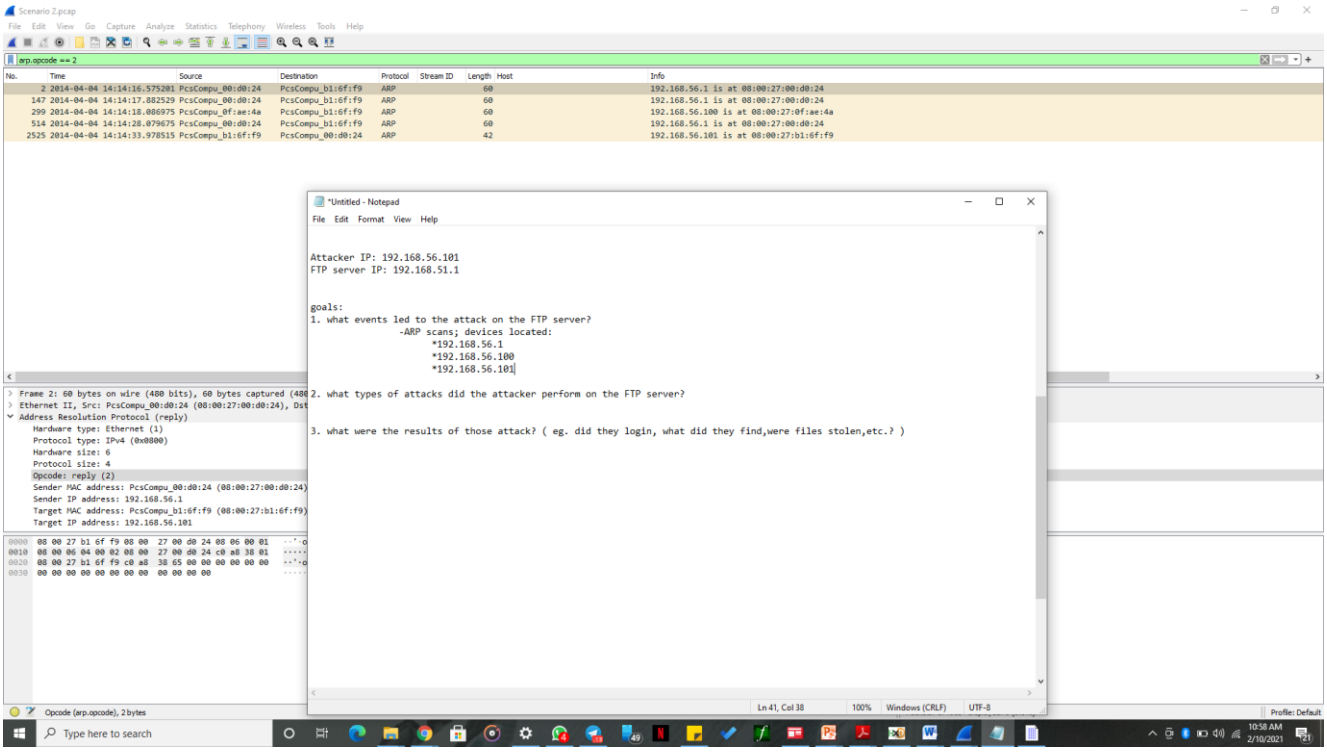


**Fig.14**

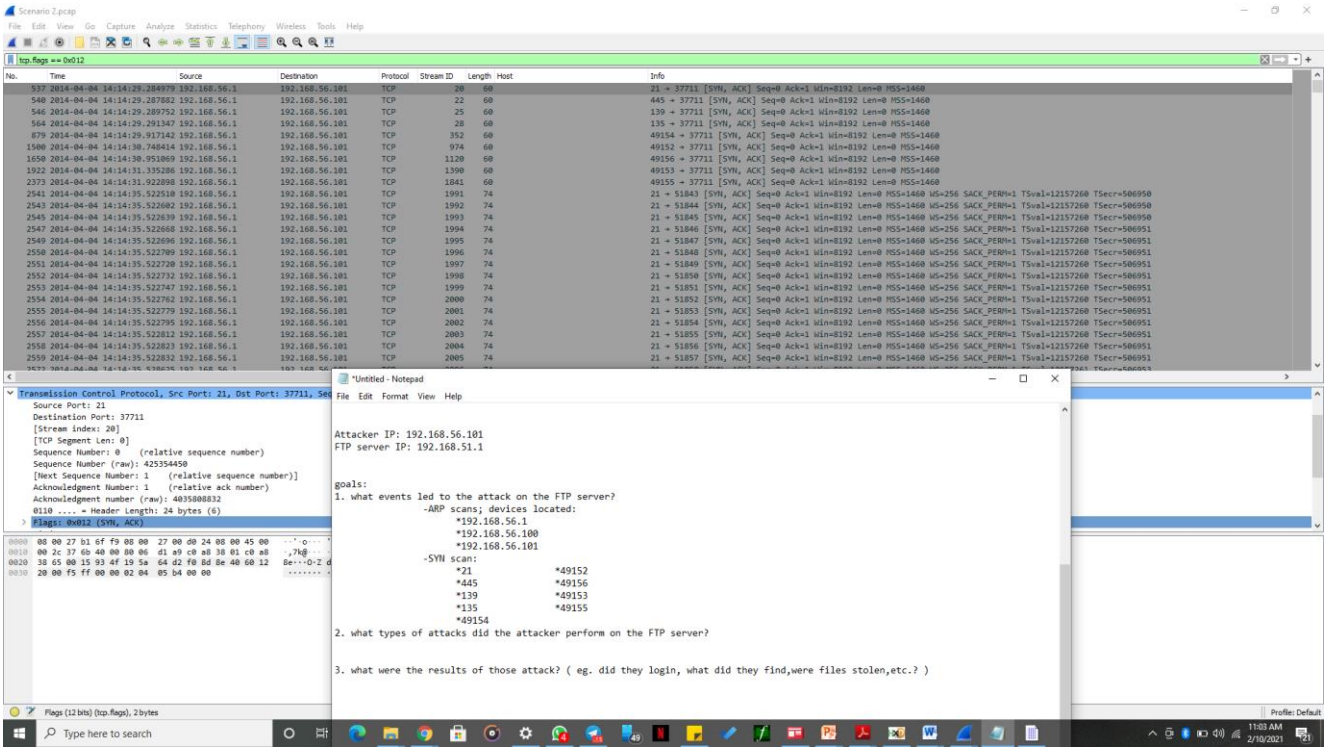
Scenario 2.pcap
*Untitled - Notepad
File Edit Format View Help
Attacker IP: 192.168.56.101
FTP server IP: 192.168.51.1
goals:
1. what events led to the attack on the FTP server?
-ARP scans; devices located:
*192.168.56.1
*192.168.56.100
*192.168.56.101
-SYN scan:
*21 *49152
*445 *49156
*139 *49153
*135 *49155
*49154
2. what types of attacks did the attacker perform on the FTP server?
3. what were the results of those attack? ( eg. did they login, what did they find,were files stolen,etc.? )

**Fig.15**

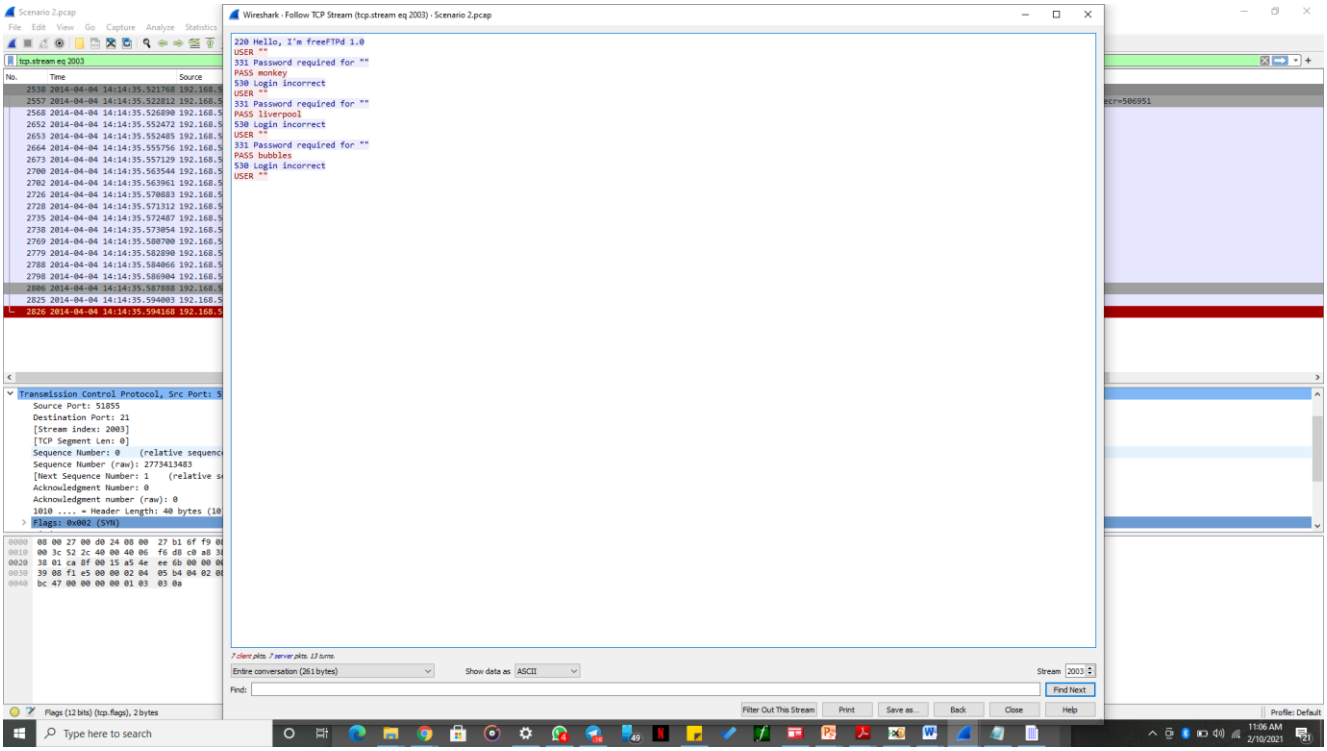
Wireshark · Follow TCP Stream (tcp.stream eq 2003) · Scenario 2.pcap
220 Hello, I'm freeFTPd 1.0
USER ""
331 Password required for ""
PASS monkey
530 Login incorrect
USER ""
331 Password required for ""
PASS liverpool
530 Login incorrect
USER ""
331 Password required for ""
PASS bubbles
530 Login incorrect
USER ""

**Fig.16**

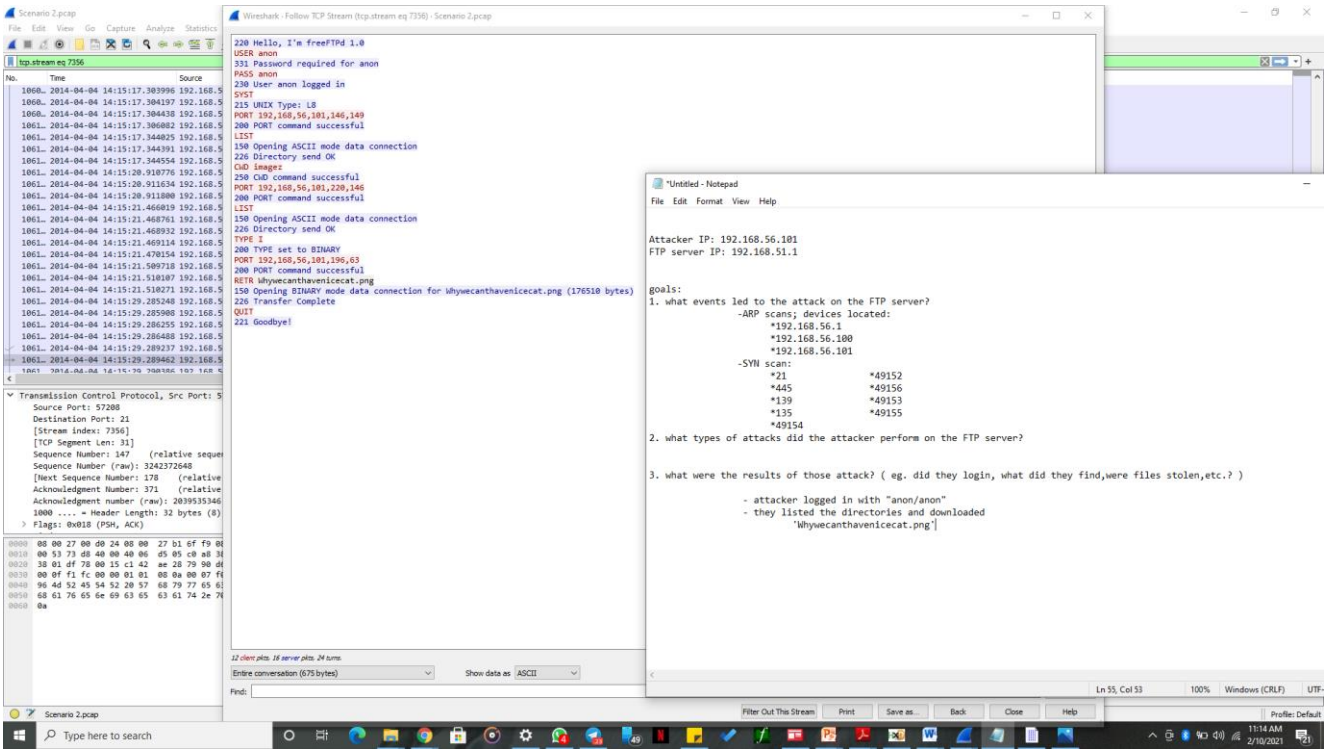
Wireshark · Follow TCP Stream (tcp.stream eq 7356) · Scenario 2.pcap
220 Hello, I'm freeFTPd 1.0
USER anon
331 Password required for anon
PASS anon
230 User anon logged in
SYST
215 UNIX Type: L8
PORT 192,168,56,101,146,149
200 PORT command successful
LIST
150 Opening ASCII mode data connection
226 Directory send OK
CWD Images
250 CWD command successful
PORT 192,168,56,101,220,146
200 PORT command successful
LIST
150 Opening ASCII mode data connection
226 Directory send OK
TYPE I
200 TYPE set to BINARY
PORT 192,168,56,101,196,63
200 PORT command successful
RETR Whywecanthavenicecat.png
150 Opening BINARY mode data connection for Whywecanthavenicecat.png (176510 bytes)
226 Transfer Complete
QUIT
221 Goodbye!
*Untitled - Notepad
File Edit Format View Help
Attacker IP: 192.168.56.101
FTP server IP: 192.168.51.1
goals:
1. what events led to the attack on the FTP server?
-ARP scans; devices located:
*192.168.56.1
*192.168.56.100
*192.168.56.101
-SYN scan:
*21 *49152
*445 *49156
*139 *49153
*135 *49155
*49154
2. what types of attacks did the attacker perform on the FTP server?
3. what were the results of those attack? ( eg. did they login, what did they find,were files stolen,etc.? )
- attacker logged in with "anon/anon"
- they listed the directories and downloaded
'Whywecanthavenicecat.png'

**Fig.17**

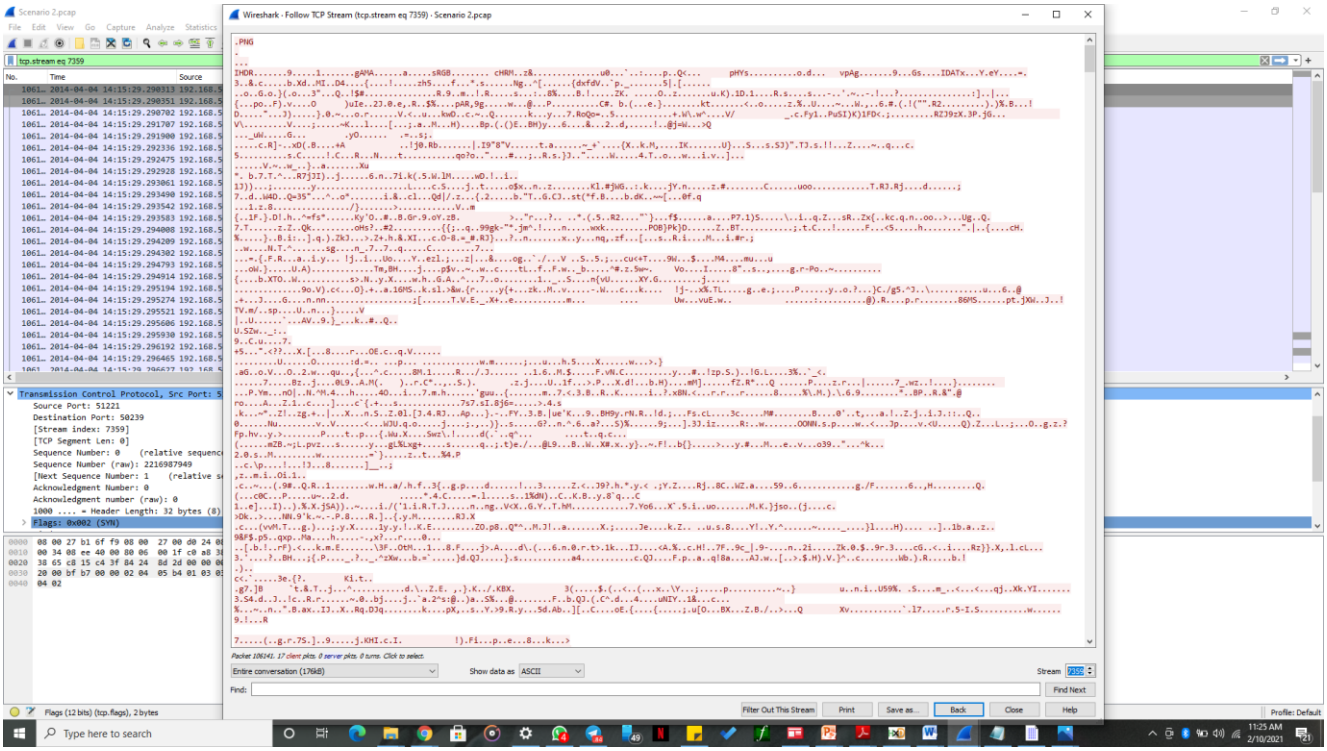
Wireshark · Follow TCP Stream (tcp.stream eq 7359) · Scenario 2.pcap

**Fig.18**

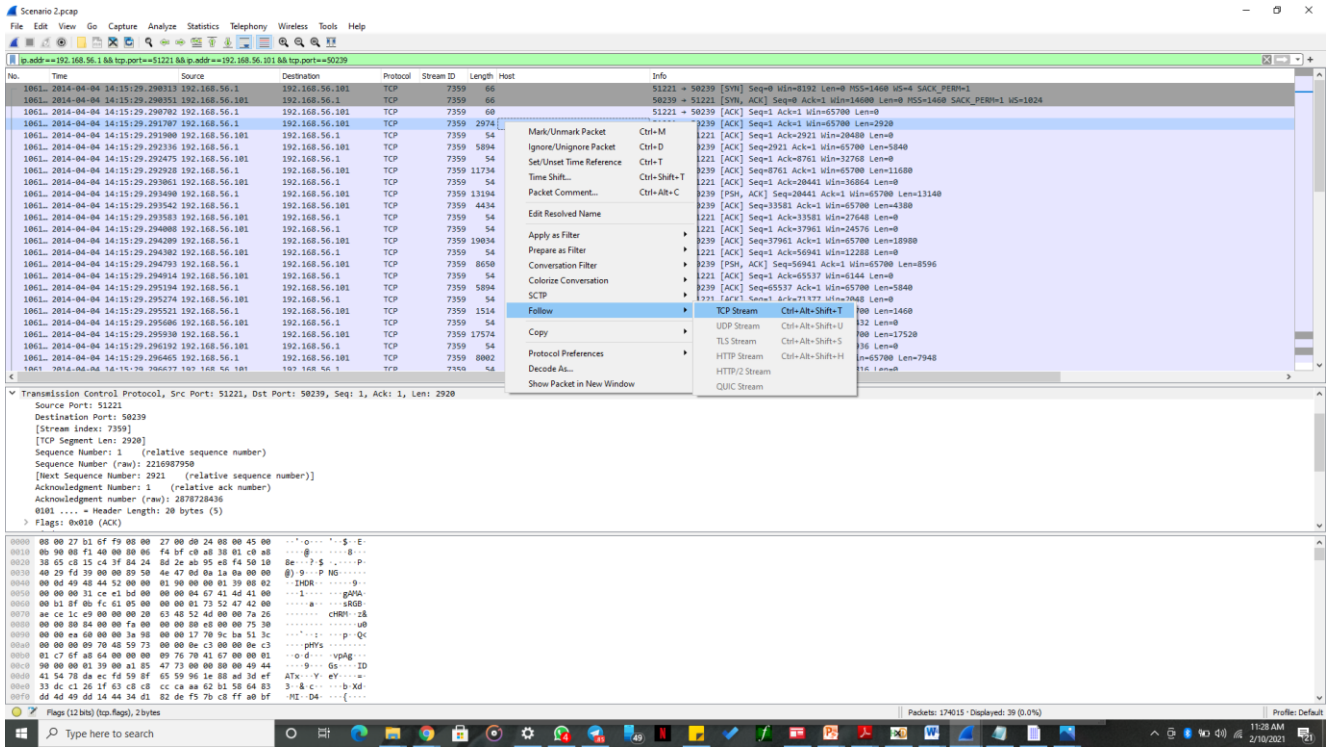
Scenario 2.pcap
File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help
Mark/Unmark Packet Ctrl+M
Ignore/Unignore Packet Ctrl+D
Set/Unset Time Reference Ctrl+T
Time Shift... Ctrl+Shift+T
Packet Comment... Ctrl+Alt+C
Edit Resolved Name
Apply as Filter
Prepare as Filter
Conversation Filter
Colorize Conversation
SCTP
Follow
Copy
Protocol Preferences
Decode As...
Show Packet in New Window
TCP Stream Ctrl+Alt+Shift+T
UDP Stream Ctrl+Alt+Shift+U
TLS Stream Ctrl+Alt+Shift+S
HTTP Stream Ctrl+Alt+Shift+H
HTTP/2 Stream
QUIC Stream
Transmission Control Protocol, Src Port: 51221, Dst Port: 50239, Seq: 1, Ack: 1, Len: 2920
Source Port: 51221
Destination Port: 50239
[Stream index: 7359]
[TCP Segment Len: 2920]
Sequence Number: 1 (relative sequence number)
Sequence Number (raw): 2216987950
[Next Sequence Number: 2921 (relative sequence number)]
Acknowledgment Number: 1 (relative ack number)
Acknowledgment number (raw): 2878728436
0101 .... = Header Length: 20 bytes (5)
Flags: 0x010 (ACK)
Type here to search
11:28 AM
2/10/2021

**Fig.19**

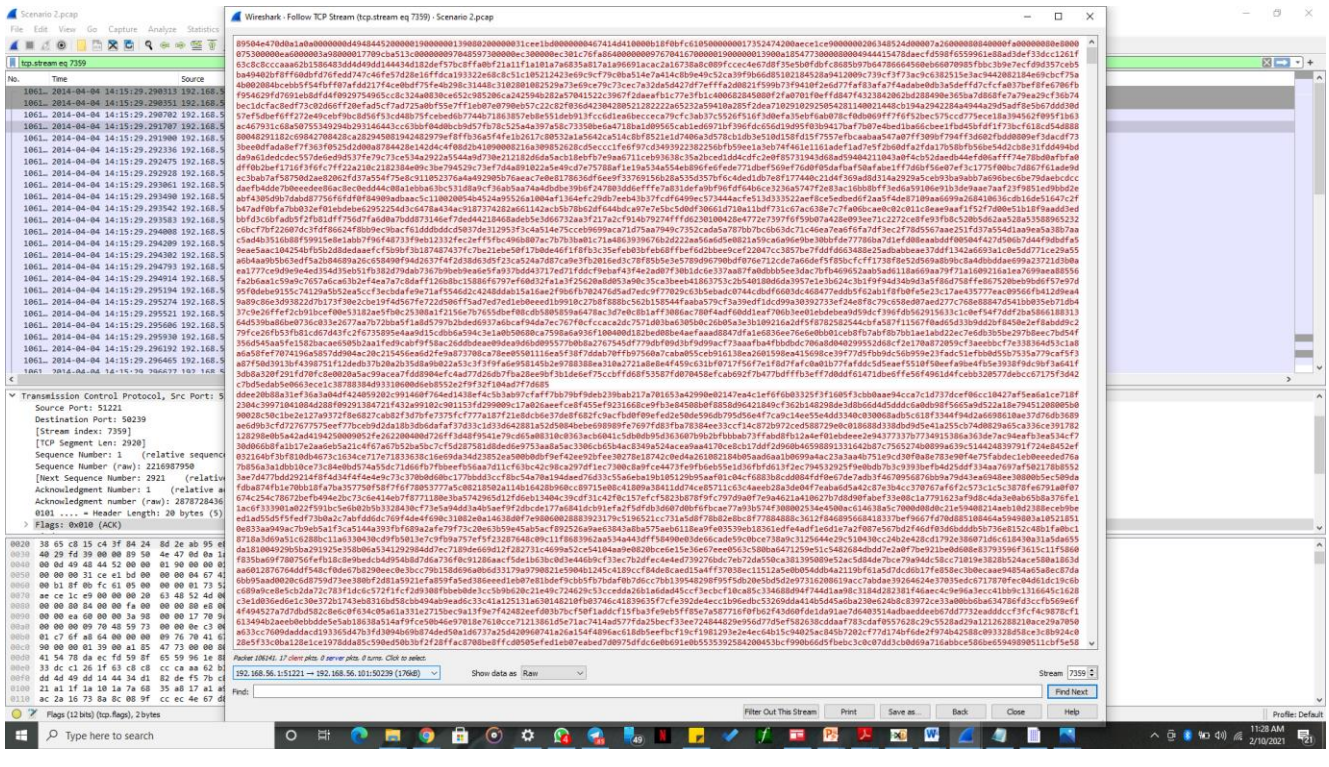
Wireshark · Follow TCP Stream (tcp.stream eq 7359) · Scenario 2.pcap
Transmission Control Protocol, Src Port: 51221, Dst Port: 50239, Seq: 1, Ack: 1, Len: 2920
Source Port: 51221
Destination Port: 50239
[Stream index: 7359]
[TCP Segment Len: 2920]
Sequence Number: 1 (relative sequence number)
Sequence Number (raw): 2216987950
[Next Sequence Number: 2921 (relative sequence number)]
Acknowledgment Number: 1 (relative ack number)
Acknowledgment number (raw): 2878728436
0101 .... = Header Length: 20 bytes (5)
Flags: 0x010 (ACK)
192.168.56.1:51221 → 192.168.56.101:50239 (176kB)
Show data as Raw
Stream 7359
Find:
Find Next
Filter Out This Stream
Print
Save as...
Back
Close
Help
Type here to search
11:28 AM
2/10/2021

**Fig.20**

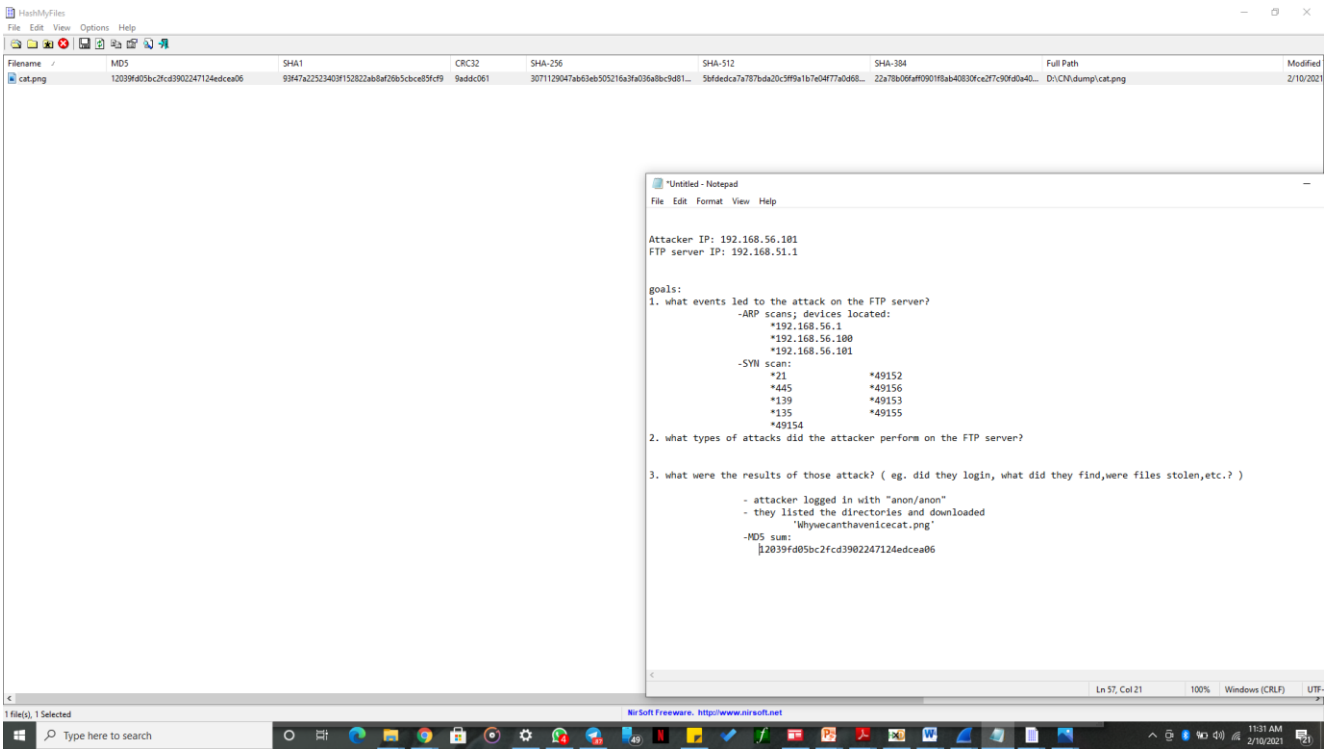
HashMyFiles
File Edit View Options Help
Filename MD5 SHA1 CRC32 SHA-256 SHA-512 SHA-384 Full Path Modified
cat.png 12039fd05bc2fcd3902247124edcea06 93f47a22523403f152822ab8af26b5cbce85fcf9 9addc061 307112904 7ab63eb505216a3fa036a8bc9d81... 5bfdedca7a787bda20c5ff9a1b7e04f77a0d68... 22a78b06faff0901f8ab40830fce2f7c90fd0a40... D:\CN\dump\cat.png 2/10/2021
1 file(s), 1 Selected
NirSoft Freeware. http://www.nirsoft.net
*Untitled - Notepad
File Edit Format View Help
Attacker IP: 192.168.56.101
FTP server IP: 192.168.51.1
goals:
1. what events led to the attack on the FTP server?
-ARP scans; devices located:
*192.168.56.1
*192.168.56.100
*192.168.56.101
-SYN scan:
*21 *49152
*445 *49156
*139 *49153
*135 *49155
*49154
2. what types of attacks did the attacker perform on the FTP server?
3. what were the results of those attack? ( eg. did they login, what did they find,were files stolen,etc.? )
- attacker logged in with "anon/anon"
- they listed the directories and downloaded
'Whywecanthavenicecat.png'
-MD5 sum:
12039fd05bc2fcd3902247124edcea06
Ln 57, Col 21 100% Windows (CRLF) UTF-
Type here to search
11:31 AM
2/10/2021

**Fig.21**

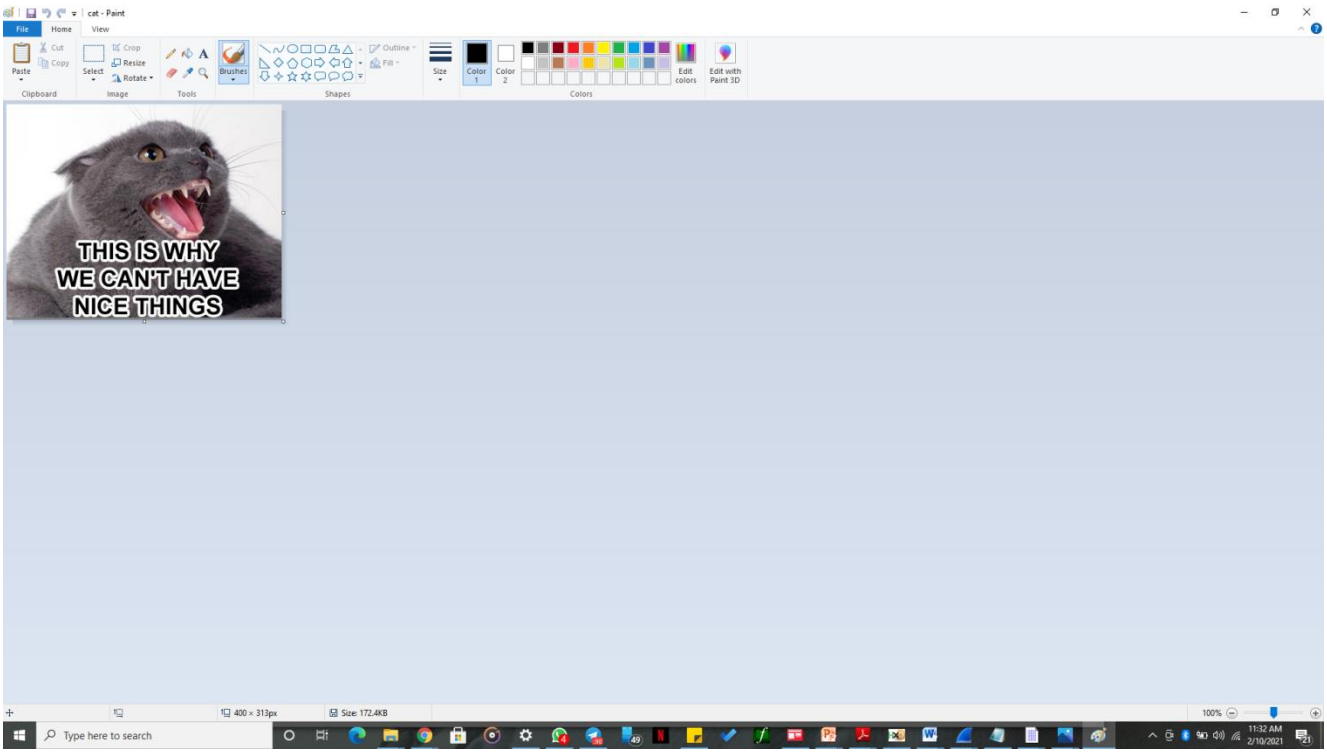
cat - Paint
File Home View
Clipboard Image Tools Shapes Colors
THIS IS WHY
WE CAN'T HAVE
NICE THINGS
400 × 313px Size: 172.4KB 100%
Type here to search
11:32 AM
2/10/2021

**Fig.22**

# CONCLUSION AND FUTURE WORKS

The cases of packet analysis demonstrated during this paper help us to understand packet analyzers, especially Wireshark are crucial to network forensics. the necessity for network packet analysis is raised by the very fact that the methods currently employed by most users for network security aren't effective enough to detect all the pc attacks, especially the newest ones as an example , antivirus software has always been the primary choice for many of enterprise and residential users. However, most antivirus software uses a signature detection method, what makes this approach inefficient for several reasons.

First, many new virus signatures are discharged yearly, and antivirus will solely discover viruses for noted valid signatures and also the unknown signatures escape the detection. Second, in line with the discussion during this paper, today's networks face threats quite virus, like malware, denial of service, port scanning covert channels, and data theft; but, antivirus computer code will solely take terribly restricted action on these varied threats. Third, hackers also can target the antivirus computer code running on a machine, resulting in multiple vulnerabilities of the system while not the notice of the user. For these totally different reasons, network traffic analysis at the packet level is critical, and it will determine many alternative threats and attacks that would stay unheeded by antivirus computer code.

In the past, packet analyzers were terribly valuable and proprietary. Wireshark has modified all that. Wireshark is one in all the simplest open supply packet analyzers obtainable these days, and it displays packet knowledge as elaborate as attainable. However, despite its wealthy toolset, it's vital to stay in mind that Wireshark isn't an intrusion detection system. Wireshark won't warn you once somebody does strange things on your network that he's not allowed to try to, and it'll not manipulate things on the network like causation packets. The quality of Wireshark is that it's a convenient and effective tool that may facilitate network security professionals discern what's very happening within the network if strange things happen, packet analysis in Wireshark will discover a broad vary of security threats and attacks against networked laptop systems.

For further scenarios to be solved, go to:

1. http://www.forensicscontest.com/puzzles
2. http://www.honeypots.org/challenges
3. http://www.malware-traffic-analysis.net/

# REFERENCES


1. *ICMP Attacks Against TCP*, RFC5927, http://tools.ietf.org/html/rfc5927.html
2. https://github.com/NetsecExplained/Advanced-Wireshark-Network-Forensics/blob/master/Scenario%202.pcap
   Source for pcap files.
3. *How to Use Filters with Wireshark*, http://www.openlogic.com/wazi/bid/188067/How-to-Use-Filters-with-Wireshark
4. https://www.researchgate.net/publication/281573989_Network_forensics_analysis_using_Wireshark
5. *ICMP Packet filtering and ICMP Attacks*, http://www.camber.com/pdf/cybersecurity/sc/Filter_ICMP_packets.pdf
6. http://csrc.nist.gov/publications/nistpubs/800-86/SP800-86.pdf
7. http://www.garykessler.net/library/file_sigs.html
8. Sample Captures, http://wiki.wireshark.org/SampleCaptures.