

	Put Student Name(s) ↓	Put Student IDs ↓	Due Date	Grade Weight
	Ishan Aakash Patel	146151238	As Posted	6%
Name	Lab 6: Analyze Network Traffic			
Instructions	<ul style="list-style-type: none"> It is an Individual assignment. Put your name + Student ID in the empty spaces above. Show your genuine signs of your work is done on your machine. This includes: <ul style="list-style-type: none"> Screenshots that show your desktop background with Date/Time. Show a pop-up bx that shows "your name + IP". Show your logged account when applicable. Optional: Your photo. Submit your report name: CYT215-Lab6-Student Name & ID 			
Challenge Scenario	You are a network forensic analyst at a medium-sized enterprise. The network team has detected unusual network activities and generated a PCAP file from one of the internal segments during a suspected attack window. Your task is to analyze the PCAP file to identify potential malicious activities, verify if any data was exfiltrated, and assess any command-and-control communications.			
Prior Knowledge	<p>Benign Traffic</p> <p>Benign traffic refers to legitimate network data that does not pose any threat to security. It includes all normal communications that occur in a network under regular operations. Understanding benign traffic is essential for network analysts because it helps establish a baseline of normal activity, making it easier to spot anomalies or malicious activities. Common examples of benign traffic include:</p> <ul style="list-style-type: none"> Web Browsing: Requests and responses over HTTP/HTTPS that are part of typical user activity. Email Communications: SMTP, POP3, and IMAP traffic used for sending and receiving emails. FTP Transfers: Normal file transfers using FTP, which might be routine backups or scheduled data transfers. DNS Queries: Regular DNS requests that resolve domain names to IP addresses, facilitating everyday internet usage. <p>Benign traffic patterns can vary widely between different networks, depending on the nature of the business and the typical activities of users. Analysts use tools like Wireshark to capture and review this traffic to understand what is typical and thereby more easily identify what is not.</p> <p>Command and Control (C&C) Communications</p> <p>Command and Control (C&C) communications refer to the signals and data passed between compromised systems and an attacker's server. These communications are a hallmark of network breaches involving malware, especially in cases of botnets or</p>			

	<p>ransomware. C&C servers issue commands to compromised systems (bots) and receive stolen data or status updates in return. Key aspects include:</p> <ul style="list-style-type: none"> • Control Mechanisms: C&C can be conducted over various protocols, including IRC, HTTP, HTTPS, or custom protocols designed to evade detection. • Purpose: These communications allow attackers to remotely manage malware, perform data exfiltration, deploy additional payloads, update configurations, or initiate denial-of-service attacks. • Detection Challenges: C&C traffic is often designed to mimic benign traffic to avoid detection by traditional security tools. For instance, using HTTPS or intermittently connecting to blend in with normal HTTPS traffic. <p>C&C communications are critical for the operational success of many malware campaigns. Detecting them involves looking for unusual outbound connections, irregular data flows, or connections to known malicious domains. Network analysts use deep packet inspection, behavior analysis, and signature-based detection to identify such communications.</p>
<p>Steps</p>	<p>Open the PCAP File:</p> <p>Open the provided <code>network_traffic.pcap</code> file in Wireshark or Network Miner. The guide below is for WireShark.</p> <p>Identify Malware Download:</p> <ul style="list-style-type: none"> • Task: Locate the HTTP GET request for <code>malware.exe</code>. • Instructions: <ul style="list-style-type: none"> ◦ Use the filter <code>http.request.method == "GET" && http.request.uri contains "malware.exe"</code>. ◦ Identify the source and destination IP addresses and note the HTTP host header. ◦ Discuss the implications of malware being downloaded over HTTP. <p>Investigate Command and Control Communication:</p> <ul style="list-style-type: none"> • Task: Find the TCP handshake followed by data suggesting C&C communication. • Instructions: <ul style="list-style-type: none"> ◦ Use the filter <code>tcp.flags.syn == 1 && tcp.port == 4444</code> to find the initial connection establishment.

	<ul style="list-style-type: none"> ○ Follow the TCP stream to view the communication. Discuss the potential signs that indicate C&C activity. <p>• Analyze Data Exfiltration:</p> <ul style="list-style-type: none"> • Task: Detect and analyze suspicious large data transfers. • Instructions: <ul style="list-style-type: none"> ○ Use the filter <code>tcp.port == 8888</code>. ○ Observe the payload size and pattern. Discuss how consistent, large payloads might indicate data exfiltration. ○ Analyze the timestamps to check if the data transfer occurred at an unusual time, suggesting malicious intent. <p>• Differentiate Benign Traffic:</p> <ul style="list-style-type: none"> • Task: Separate and identify benign DNS and HTTP traffic. • Instructions: <ul style="list-style-type: none"> ○ For DNS: Use the filter <code>udp.port == 53</code>. ○ For HTTP: Use the filter <code>http.request.method == "GET" && http.host == "www.example.com"</code>. ○ Discuss the characteristics of benign traffic and how it differs from the malicious traffic observed. <p>• Reporting:</p> <ul style="list-style-type: none"> • Task: Prepare a forensic report detailing the findings. • Instructions: <ul style="list-style-type: none"> ○ Summarize the identified malicious and benign activities. ○ Provide detailed evidence for each activity (screenshots, Wireshark filters used, etc.). ○ Recommend actions based on the findings.
Grading Alerts	<ul style="list-style-type: none"> • If you do NOT use this template or delete any part of it or use any other template, you will be degraded. • If you do NOT follow the file naming convention, you will be degraded. • If you do NOT submit your file in PDF; you will be degraded. • If you do NOT show your account real name (when applicable); you will be degraded. • If you do NOT show your machine desktop background (with date & time) and IP, you will be degraded. • If you do NOT write (in your own words) your learning experience for the activity practices, you will be degraded.

Wireshark

The screenshot shows the Wireshark Network Analyzer interface within a Kali Linux virtual machine. The interface is divided into several sections:

- Library:** A sidebar on the left showing a list of virtual machines, including "My Computer", "Metasploitable2-Linux", "Metasploitable", "Windows 10", "Lab-11", "SIFT", "Mobisec x64 2.0.2", "Android", and "Kali".
- File Menu:** The top menu bar includes "File", "Edit", "View", "Go", "Capture", "Analyze", "Statistics", "Telephony", "Wireless", "Tools", and "Help".
- Display Filter:** A bar below the menu bar with the text "Apply a display filter ... <Ctrl-/>".
- Welcome to Wireshark:** A central panel with a "Welcome to Wireshark" message, an "Open" button, and a file path "/home/kali/capture.pcap (2770 KB)".
- Capture:** A section below the welcome message showing a list of network interfaces. The "eth0" interface is selected and highlighted in blue. Other interfaces listed include "any", "Loopback: lo", "bluetooth-monitor", "nflag", "nfqueue", "dbus-system", "dbus-session", and "Cisco remote capture: ciscodump".
- Learn:** A section at the bottom of the welcome panel with links to "User's Guide", "Wiki", "Questions and Answers", "Mailing Lists", "SharkFest", "Wireshark Discord", and "Donate". It also states "You are running Wireshark 4.2.5 (Git v4.2.5 packaged as 4.2.5-1)".
- Status Bar:** At the bottom of the window, it shows "Ready to load or capture", "No Packets", and "Profile: Default".

At the bottom of the screen, there is a Windows taskbar with a search bar, several application icons, and a system tray showing the date and time as "16:15 7/17/2024".

Opening the pcap file in wireshark

The screenshot shows a Kali Linux virtual machine running Wireshark. The main window displays the network traffic file `network_traffic.pcap`. The packet list shows five packets, with the first packet selected:

No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000	192.168.0.2	192.168.0.1	HTTP	97	GET /malware.exe HTTP/1.1
2	0.000483	192.168.0.2	192.168.0.1	TCP	50	54321 → 4444 [SYN] Seq=0 Win=8192 Len=10
3	0.000817	192.168.0.2	192.168.0.1	TCP	558	12345 → 8888 [SYN] Seq=0 Win=8192 Len=518
4	0.001126	192.168.0.2	8.8.8.8	DNS	61	Standard query 0x0000 A www.example.com
5	0.001722	192.168.0.2	192.168.0.1	HTTP	81	GET / HTTP/1.1

The details pane shows the structure of the selected frame:

- Frame 1: 97 bytes on wire (776 bits), 97 bytes captured (776 bits)
- Internet Protocol Version 4, Src: 192.168.0.2, Dst: 192.168.0.1
- Transmission Control Protocol, Src Port: 12345, Dst Port: 80, Seq: 0, Len: 57
- Hypertext Transfer Protocol

A small window in the foreground displays the following information:

```
Name: Ishan Aakash Patel
StudentID: 146151238
```

The bottom of the screen shows the Windows taskbar with the date and time: 16:16, 7/17/2024.

Task 1 : Identify Malware Download

The screenshot displays a Kali Linux virtual machine running VMware Workstation. The main window shows a network traffic capture file named `network_traffic.pcap`. The filter applied is `http.request.method == "GET" && http.request.uri contains "malware.exe"`. A single packet is listed in the packet list pane:

No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000	192.168.0.2	192.168.0.1	HTTP	97	GET /malware.exe HTTP/1.1

The packet details pane shows the following structure:

- Frame 1: 97 bytes on wire (776 bits), 97 bytes captured (776 bits)
- Internet Protocol Version 4, Src: 192.168.0.2, Dst: 192.168.0.1
- Transmission Control Protocol, Src Port: 12345, Dst Port: 80, Seq: 0, Len: 57
- Hypertext Transfer Protocol
 - GET /malware.exe HTTP/1.1\r\n
 - Host: malicious-server.com\r\n
 - [Full request URI: http://malicious-server.com/malware.exe]
 - [HTTP request 1/1]

The raw data pane shows the hex and ASCII representation of the packet. A red box highlights the GET request line and the host information. A small window is open in the foreground with the following text:

```
File Edit View
Name:
Name: Ishan Aakash Patel
StudentID: 146151238
```

The bottom status bar indicates the current position is at line 2, column 21, with 45 characters displayed at 100% zoom in a window titled `network_traffic.pcap`. The system tray shows the date and time as 16:21 on 7/17/2024.

Source IP: 192.168.0.2 Destination IP: 192.168.0.1 HTTP Host header: malicious-server.com

Implications of malware being downloaded over HTTP:

1. Lack of encryption: HTTP transfers data in plaintext, allowing potential attackers to intercept and view the content, including sensitive information like usernames, passwords, or in this case, malicious code.
2. No data integrity: Without HTTPS, there's no way to verify if the content has been tampered with in transit, potentially allowing man-in-the-middle attacks.
3. Easier detection: Network administrators and security tools can more easily detect and block suspicious HTTP traffic compared to encrypted HTTPS traffic.
4. Vulnerability to DNS hijacking: Attackers could potentially redirect HTTP requests to their own servers more easily than with HTTPS.
5. Lack of server authentication: There's no way to verify if the server is actually the intended one, increasing the risk of connecting to malicious servers.
6. Potential for network-level attacks: Unencrypted traffic is more susceptible to various network-level attacks and manipulations.
7. Non-compliance: Many security standards and regulations require the use of encryption for data transfer, making HTTP downloads of sensitive content non-compliant.

Task 2 : Investigation Command and Control Communication

Kali - VMware Workstation

File Edit View VM Tabs Help

Library

My Computer

- Metasploitable2-Linux
- Metasploitable
- Windows 10
- Lab-11
- SIFT
- Mobisec x64 2.0.2
- Android
- Kali

network_traffic.pcap

File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help

tcp.stream eq 1

No.	Time	Source	Destination	Protocol	Length	Info
2	0.000483	192.168.0.2	192.168.0.1	TCP	50	54321 → 4444 [SYN] Seq=0 Win=8192 Len=10

Destination Port: 4444
[Stream index: 1]
[Conversation completeness: Incomplete (9)]
[TCP Segment Len: 10]
Sequence Number: 0 (relative sequence number)
Sequence Number (raw): 0
[Next Sequence Number: 11 (relative sequence number)]
Acknowledgment Number: 0
Acknowledgment number (raw): 0
0101 = Header Length: 20 bytes (5)
Flags: 0x002 (SYN)
Window: 8192
[Calculated window size: 8192]
Checksum: 0xb294 [unverified]
[Checksum Status: Unverified]
Urgent Pointer: 0
[Timestamps]
[SEQ/ACK analysis]
TCP payload (10 bytes)
Data (10 bytes)
Data (data), 10 bytes

Wireshark - Follow TCP Stream (tcp.stream eq 1) - network_traffic.pcap

Hello, C2!

Name: Ishan Aakash Patel
StudentID: 146151238

Ln 2, Col 21 45 characters 100% Window UTF-8

To direct input to this VM, move the mouse pointer inside or press Ctrl+G.

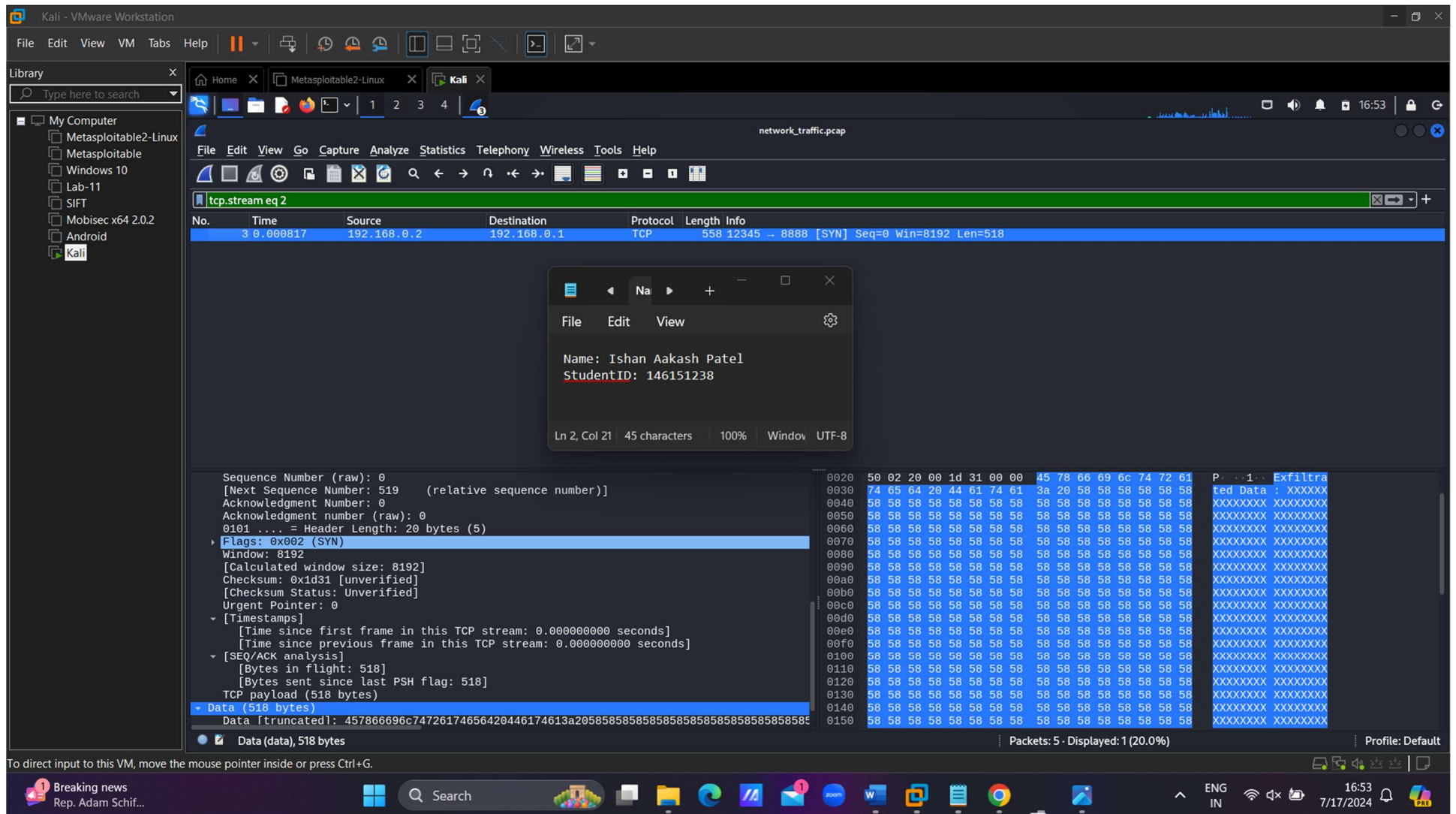
Breaking news
Rep. Adam Schiff...

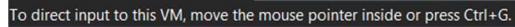
Search

ENG IN 16:36 7/17/2024

1. Use of port 4444: This is a non-standard port often associated with malware and backdoors.
2. Short, simple initial message: The payload "Hello, C2!" suggests a basic check-in or beacon to a command server.
3. TCP stream content: The presence of a simple greeting could be an initial handshake or identification message to the C&C server.
4. Unusual source/destination: The communication is between local IP addresses (192.168.0.2 to 192.168.0.1), which could indicate an infected internal machine contacting a compromised server or pivot point.
5. SYN flag: The TCP SYN flag indicates the start of a new connection, potentially for ongoing communication with the C&C server.
6. Small payload size: The 10-byte payload is consistent with a minimal beacon or command acknowledgment.
7. Lack of standard application data: There's no indication of normal application traffic, suggesting this could be malware communication.

Task 3 : Analyze Data Exfiltration





Payload size and pattern:

- The TCP stream shows a large payload of 518 bytes.
- The data appears to be consistent and repetitive, with many "58" byte values visible in the hex dump.
- This pattern of large, consistent payloads could indicate data exfiltration. Attackers often compress or encode stolen data before transmission, which can result in uniform, seemingly random data patterns.
- The use of port 8888, which is non-standard, further raises suspicion.

Timestamps and timing:

- The timestamp shown in the capture is 3.000817 seconds from the start of the capture.
- Without more context about the normal operating hours of the system or network, it's difficult to definitively state if this is an unusual time.
- However, data exfiltration often occurs during off-hours to avoid detection. The fact that this large data transfer is happening might be suspicious depending on the expected network behavior.

Additional observations:

- The communication is between internal IP addresses (192.168.0.2 to 192.168.0.1), which could indicate an already compromised internal system acting as a staging point for data exfiltration.
- The SYN flag is set, suggesting the start of a new connection for this data transfer.

Task 4 : Differentiate Benign Traffic

For DNS

The screenshot displays a Kali Linux virtual machine running VMware Workstation. The main application window is Wireshark, which is analyzing a network traffic capture file named 'network_traffic.pcap'. A filter is applied to the packet list: 'udp.port == 53'. The packet list shows a single packet (No. 4) at time 0.001126, sourced from 192.168.0.2 and destined for 8.8.8.8, identified as a DNS Standard query. The packet details pane on the left shows the structure of the DNS query, including the transaction ID (0x0000), flags (0x0100), and a single query for 'www.example.com' of type A and class IN. The packet bytes pane on the right shows the raw data in hexadecimal and ASCII. A small text editor window is open in the foreground, displaying the text: 'Name: Ishan Aakash Patel' and 'StudentID: 146151238'. The bottom status bar of the VM shows the system clock as 17:01 on 7/17/2024.

Kali - VMware Workstation

File Edit View VM Tabs Help

Library

My Computer

- Metasploitable2-Linux
- Metasploitable
- Windows 10
- Lab-11
- SIFT
- Mobisec x64 2.0.2
- Android
- Kali

network_traffic.pcap

File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help

Filter: udp.port == 53

No.	Time	Source	Destination	Protocol	Length	Info
4	0.001126	192.168.0.2	8.8.8.8	DNS	61	Standard query 0x0000 A www.example.com

Frame 4: 61 bytes on wire (488 bits), 61 bytes captured (488 bits) on interface
Internet Protocol Version 4, Src: 192.168.0.2, Dst: 8.8.8.8
User Datagram Protocol, Src Port: 33333, Dst Port: 53
Domain Name System (query)
Transaction ID: 0x0000
Flags: 0x0100 Standard query
Questions: 1
Answer RRs: 0
Authority RRs: 0
Additional RRs: 0
Queries
www.example.com: type A, class IN
Name: www.example.com
[Name Length: 15]
[Label Count: 3]
Type: A (1) (Host Address)
Class: IN (0x0001)

Text item (text), 21 bytes

File Edit View

Name: Ishan Aakash Patel
StudentID: 146151238

Ln 2, Col 21 | 45 characters | 100% | Window UTF-8

Packets: 5 - Displayed: 1 (20.0%)

Profile: Default

To direct input to this VM, move the mouse pointer inside or press Ctrl+G.

81°F Mostly cloudy

Search

ENG IN 17:01 7/17/2024

For HTTP

Kali - VMware Workstation

File Edit View VM Tabs Help

Library

- My Computer
 - Metasploitable2-Linux
 - Metasploitable
 - Windows 10
 - Lab-11
 - SIFT
 - Mobisec x64 2.0.2
 - Android
 - Kali

network_traffic.pcap

File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help

http.request.method == "GET" && http.host == "www.example.com"

No.	Time	Source	Destination	Protocol	Length	Info
5	0.001722	192.168.0.2	192.168.0.1	HTTP	81	GET / HTTP/1.1

Ln 2, Col 21 | 45 characters | 100% | Window | UTF-8

Name: Ishan Aakash Patel
StudentID: 146151238

Frame 5: 81 bytes on wire (648 bits), 81 bytes captured (648 bits)
Internet Protocol Version 4, Src: 192.168.0.2, Dst: 192.168.0.1
Transmission Control Protocol, Src Port: 23456, Dst Port: 80, Seq: 0, Len: 41
Hypertext Transfer Protocol
GET / HTTP/1.1\r\n
[Expert Info (Chat/Sequence): GET / HTTP/1.1\r\n]
[GET / HTTP/1.1\r\n]
[Severity level: Chat]
[Group: Sequence]
Request Method: GET
Request URI: /
Request Version: HTTP/1.1
Host: www.example.com\r\n\r\n[Full request URI: http://www.example.com/]
[HTTP request 1/1]

0000 45 00 00 51 00 01 00 00 40 06 f9 52 c0 a8 00 02 E: Q...@...R...
0010 c0 a8 00 01 5b a0 00 50 00 00 00 00 00 00 00 00 ...[...P...
0020 50 02 20 00 7b 2a 00 00 47 45 54 20 2f 20 48 54 P...{*...GET / H
0030 54 50 2f 31 2e 31 0d 0a 48 6f 73 74 3a 20 77 77 TP/1.1... Host: ww
0040 77 2e 65 78 61 6d 70 6c 65 2e 63 6f 6d 0d 0a 0d w.exempl e.com...
0050 0a

HTTP Request HTTP-Version (http.request.version), 8 bytes

Packets: 5 - Displayed: 1 (20.0%)

Profile: Default

To direct input to this VM, move the mouse pointer inside or press Ctrl+G.

Upcoming Earnings

Search

ENG IN

17:03 7/17/2024

Characteristics of benign traffic:

1. Standard protocols and ports: The DNS query uses UDP port 53, which is the standard port for DNS.
2. Expected destinations: The DNS query is sent to 8.8.8.8, a well-known public DNS server (Google's).
3. Normal query content: The DNS request is for "www.example.com", a common placeholder domain often used in documentation and testing.
4. Appropriate packet size: The DNS query is 61 bytes, which is a typical size for a standard DNS request.
5. Clear, unobfuscated data: The domain being queried is visible in plaintext, not encoded or obfuscated.
6. Expected behavior: A single DNS query for a domain name is normal network behavior.
7. Standard flags: The packet shows a "Standard query" flag, which is expected for normal DNS traffic.

Differences from malicious traffic:

1. No suspicious ports: Unlike the earlier observed traffic on ports 4444 and 8888, this uses a standard port.
2. Public destination: The DNS query goes to a public IP, not an internal address like the suspicious traffic.
3. No large data transfers: This is a small DNS query, unlike the large, repetitive data seen in potential exfiltration.
4. Expected protocol behavior: This follows standard DNS protocol, unlike potential C2 communications seen earlier.
5. No encoded payloads: The earlier suspicious traffic had repetitive, possibly encoded data. This DNS query is clear and understandable.
6. Legitimate domain: "www.example.com" is a known, safe domain, unlike potential malicious domains or IP addresses seen in attack traffic.
7. No signs of evasion: There are no attempts to hide the nature of this traffic, unlike malware which often tries to blend in or obfuscate its communications.

Learning Experience

This lab was really eye-opening for me. I got to use Wireshark to look at real network traffic and figure out what was normal and what wasn't. It was like being a detective, searching for clues in all the data going back and forth. I learned how to spot things that didn't look right, like malware downloads and suspicious connections. It was surprising to see how attackers try to hide their activities by making them look like normal traffic.

The most interesting part was seeing how different types of traffic look in Wireshark. Normal stuff like DNS queries and web browsing has certain patterns. But when there's something fishy going on, like data being stolen or malware talking to its control server, it stands out if you know what to look for. This lab made me realize how important it is to understand normal network behavior so you can catch the bad stuff. I feel like I've gained some real-world skills that could be useful in a cybersecurity job.