Wireshark LAB 3.1 Incident Report Cyber Attack (V1.1)

OBJECTIVES:

1. Use Wireshark to identify a security issue.
2. You will create an Incident Report detailing what happened (issues).

REQUIREMENTS:

- ☐ Wireshark Application
- ☐ OS (Windows, macOS, or Linux)

STEPS:

Part 1 - Analyzing HTTP Traffic.
Part 2 - What is HTTP Basic Authentication?
Part 3 - Incident Reports.

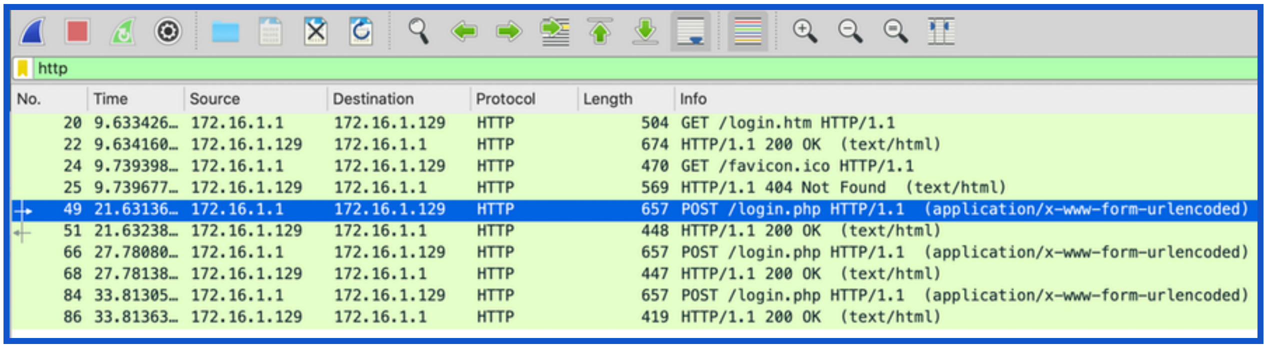
Part 1- Analyzing HTTP Traffic:

Similar to the Telnet and FTP protocols, HTTP also sends information over a network without encryption (in plain text). To show how a threat actor can find this information, we will analyze a PCAP file to find a password.

Download this file and double-click it to open it in Wireshark: http login.pcag. This file will show the HTTP connection made when visiting certain websites.

Once the file is open in Wireshark, you can move to the next step.

Finding an HTTP Password



http

No.	Time	Source	Destination	Protocol	Length	Info
20	9.633426…	172.16.1.1	172.16.1.129	HTTP	504	GET /login.htm HTTP/1.1
22	9.634160…	172.16.1.129	172.16.1.1	HTTP	674	HTTP/1.1 200 OK (text/html)
24	9.739398…	172.16.1.1	172.16.1.129	HTTP	470	GET /favicon.ico HTTP/1.1
25	9.739677…	172.16.1.129	172.16.1.1	HTTP	569	HTTP/1.1 404 Not Found (text/html)
49	21.63136…	172.16.1.1	172.16.1.129	HTTP	657	POST /login.php HTTP/1.1 (application/x-www-form-urlencoded)
51	21.63238…	172.16.1.129	172.16.1.1	HTTP	448	HTTP/1.1 200 OK (text/html)
66	27.78080…	172.16.1.1	172.16.1.129	HTTP	657	POST /login.php HTTP/1.1 (application/x-www-form-urlencoded)
68	27.78138…	172.16.1.129	172.16.1.1	HTTP	447	HTTP/1.1 200 OK (text/html)
84	33.81305…	172.16.1.1	172.16.1.129	HTTP	657	POST /login.php HTTP/1.1 (application/x-www-form-urlencoded)
86	33.81363…	172.16.1.129	172.16.1.1	HTTP	419	HTTP/1.1 200 OK (text/html)

Now that the file is open on Wireshark, follow the steps below to find the user's username and password:

1. In Wireshark, in the “Apply a display filter” box (top), type HTTP and press the **Enter** key. Wireshark filters the packets, showing only the packets using HTTP.
2. In the Packet List (top pane), in the “Info” column, find the first POST request and click it. A POST request is used to send data to a server to create or update a resource.

In the Packet Details (middle pane), you will find the username and password of this user in either the "Hypertext Transfer Protocol" or the “HTML Form URL Encoded” container by clicking on the gray arrow to expand each section.

Questions:

Based on the evidence you found in the middle pane of Wireshark, what is the possible username and password for this user? How do you know?

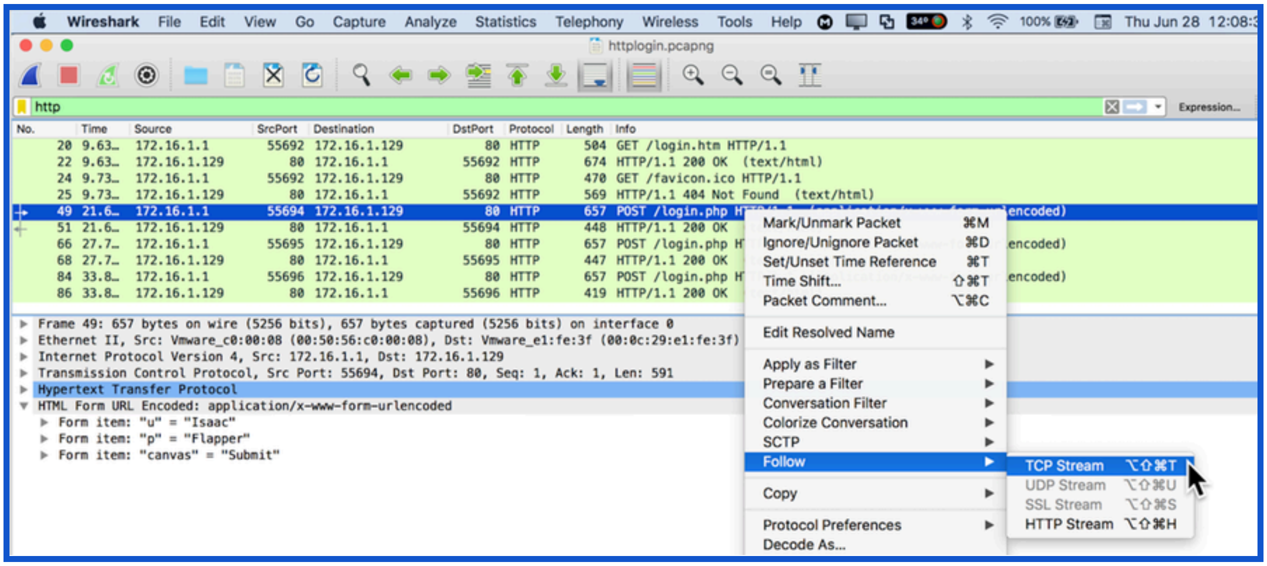
Username : Isaac

Password: Flapper

It was written under the HTML Form URL Encoded container and can be seen by expanding it.

Exchanging information from one computer to another is like a conversation. A client (end-user on a smartphone, laptop, etc.) speaks to a **server** (a computer that hosts web pages or an app) to ask for and give information. We will use Wireshark to follow this conversation, which is called a TCP Stream.

In Wireshark, go to the top pane, and look in the "Info" column. Right-click the first **POST** request, and then click **Follow**, and **TCP Stream** as shown below:



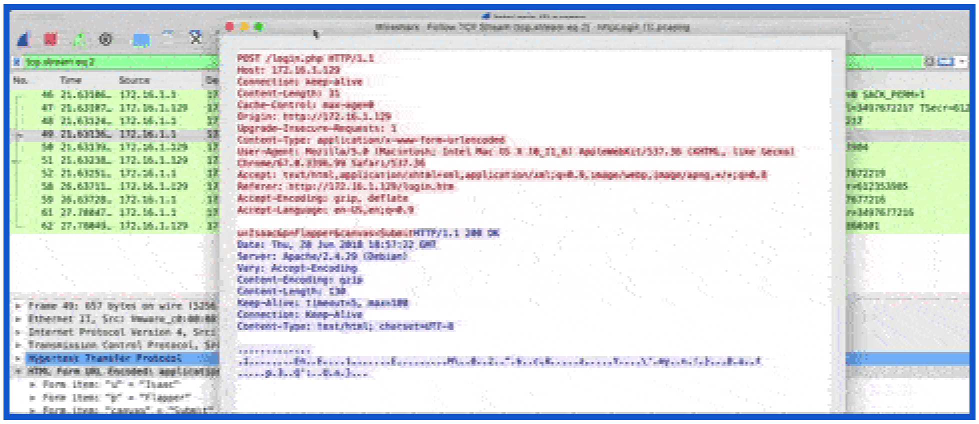
```
No.  Time   Source        SrcPort Destination   DstPort Protocol Length Info
  20 9.63…  172.16.1.1      55692 172.16.1.129       80 HTTP        504 GET /login.htm HTTP/1.1
  22 9.63…  172.16.1.129       80 172.16.1.1      55692 HTTP        674 HTTP/1.1 200 OK  (text/html)
  24 9.73…  172.16.1.1      55692 172.16.1.129       80 HTTP        470 GET /favicon.ico HTTP/1.1
  25 9.73…  172.16.1.129       80 172.16.1.1      55692 HTTP        569 HTTP/1.1 404 Not Found  (text/html)
  49 21.6…  172.16.1.1      55694 172.16.1.129       80 HTTP        657 POST /login.php H                encoded)
  51 21.6…  172.16.1.129       80 172.16.1.1      55694 HTTP        448 HTTP/1.1 200 OK
  66 27.7…  172.16.1.1      55695 172.16.1.129       80 HTTP        657 POST /login.php H                encoded)
  68 27.7…  172.16.1.129       80 172.16.1.1      55695 HTTP        447 HTTP/1.1 200 OK
  84 33.8…  172.16.1.1      55696 172.16.1.129       80 HTTP        657 POST /login.php H                encoded)
  86 33.8…  172.16.1.129       80 172.16.1.1      55696 HTTP        419 HTTP/1.1 200 OK

▶ Frame 49: 657 bytes on wire (5256 bits), 657 bytes captured (5256 bits) on interface 0
▶ Ethernet II, Src: Vmware_c0:00:08 (00:50:56:c0:00:08), Dst: Vmware_e1:fe:3f (00:0c:29:e1:fe:3f)
▶ Internet Protocol Version 4, Src: 172.16.1.1, Dst: 172.16.1.129
▶ Transmission Control Protocol, Src Port: 55694, Dst Port: 80, Seq: 1, Ack: 1, Len: 591
▶ Hypertext Transfer Protocol
▼ HTML Form URL Encoded: application/x-www-form-urlencoded
   ▶ Form item: "u" = "Isaac"
   ▶ Form item: "p" = "Flapper"
   ▶ Form item: "canvas" = "Submit"
```

A new pop-up window should appear that shows the client's request in red and the server's response in blue.

What information can you see in the TCP Stream that can be sensitive or confidential?

In this step, we are going to learn a bit more about what the user typed in by viewing the HTTP reply. But first, we must restore the packet filter to "HTTP." Follow the three steps below (also seen in the gif) to do this]:

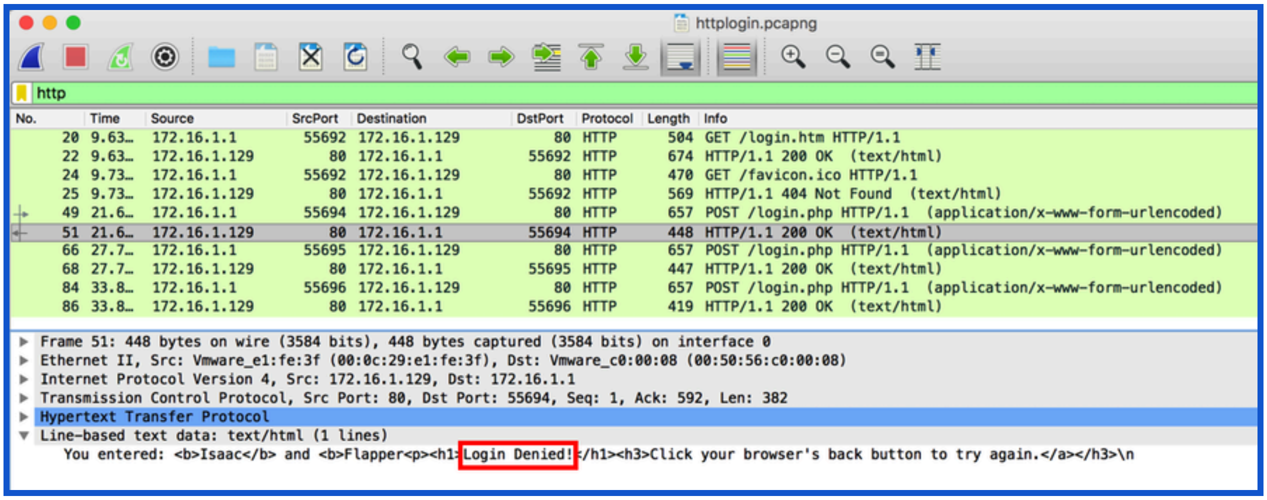
1. Close the "**Follow TCP Stream**" box.
2. In Wireshark, at the top in the "Apply a display filter" box on the right side, click the **X** to clear the filter.
3. In the "Apply a display filter" box, type http and press the **Enter** key. All of the "http" packets will appear.


Wireshark · Follow TCP Stream (tcp.stream eq 2) · httplogin.pcapng

```
POST /login.php HTTP/1.1
Host: 172.16.1.129
Connection: keep-alive
Content-Length: 31
Cache-Control: max-age=0
Origin: http://172.16.1.129
Upgrade-Insecure-Requests: 1
Content-Type: application/x-www-form-urlencoded
User-Agent: Mozilla/5.0 (Macintosh; Intel Mac OS X 10_11_6) AppleWebKit/537.36 (KHTML, like Gecko)
Chrome/67.0.3396.99 Safari/537.36
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,image/apng,*/*;q=0.8
Referer: http://172.16.1.129/login.htm
Accept-Encoding: gzip, deflate
Accept-Language: en-US,en;q=0.9

u=Isaac&p=Flapper&canvas=SubmitHTTP/1.1 200 OK
Date: Thu, 28 Jun 2018 18:57:22 GMT
Server: Apache/2.4.29 (Debian)
Vary: Accept-Encoding
Content-Encoding: gzip
Content-Length: 130
Keep-Alive: timeout=5, max=100
Connection: Keep-Alive
Content-Type: text/html; charset=UTF-8
```

Now that you are looking at the HTTP display filter again, follow the steps below to view the HTTP reply:

1. In the Packet List (top pane), under the “Info” column, click the packet after the first **POST** request, which is No. 51. The info column is labeled “**HTTP/1.1 200 OK**” (length is 448).
2. In the Packet Details (middle pane), expand (open) the "**Line-based test data**" container. The server's reply is now readable, saying “**Login Denied!**”
3. Perform the same steps, and look at this information for Packets No. 68 and 86.



http

No.	Time	Source	SrcPort	Destination	DstPort	Protocol	Length	Info
20	9.63…	172.16.1.1	55692	172.16.1.129	80	HTTP	504	GET /login.htm HTTP/1.1
22	9.63…	172.16.1.129	80	172.16.1.1	55692	HTTP	674	HTTP/1.1 200 OK (text/html)
24	9.73…	172.16.1.1	55692	172.16.1.129	80	HTTP	470	GET /favicon.ico HTTP/1.1
25	9.73…	172.16.1.129	80	172.16.1.1	55692	HTTP	569	HTTP/1.1 404 Not Found (text/html)
49	21.6…	172.16.1.1	55694	172.16.1.129	80	HTTP	657	POST /login.php HTTP/1.1 (application/x-www-form-urlencoded)
51	21.6…	172.16.1.129	80	172.16.1.1	55694	HTTP	448	HTTP/1.1 200 OK (text/html)
66	27.7…	172.16.1.1	55695	172.16.1.129	80	HTTP	657	POST /login.php HTTP/1.1 (application/x-www-form-urlencoded)
68	27.7…	172.16.1.129	80	172.16.1.1	55695	HTTP	447	HTTP/1.1 200 OK (text/html)
84	33.8…	172.16.1.1	55696	172.16.1.129	80	HTTP	657	POST /login.php HTTP/1.1 (application/x-www-form-urlencoded)
86	33.8…	172.16.1.129	80	172.16.1.1	55696	HTTP	419	HTTP/1.1 200 OK (text/html)

```
▶ Frame 51: 448 bytes on wire (3584 bits), 448 bytes captured (3584 bits) on interface 0
▶ Ethernet II, Src: Vmware_e1:fe:3f (00:0c:29:e1:fe:3f), Dst: Vmware_c0:00:08 (00:50:56:c0:00:08)
▶ Internet Protocol Version 4, Src: 172.16.1.129, Dst: 172.16.1.1
▶ Transmission Control Protocol, Src Port: 80, Dst Port: 55694, Seq: 1, Ack: 592, Len: 382
▶ Hypertext Transfer Protocol
▼ Line-based text data: text/html (1 lines)
    You entered: <b>Isaac</b> and <b>Flapper<p><h1>Login Denied!</h1><h3>Click your browser's back button to try again.</a></h3>\n
```

Questions:

☞ **Isaac attempted to login three times (packet no. 51,68 and 86). Why did he try so many times? What happened on the final attempt?**

Issac attempted so many times because the login was denied. On the final attempt the login was approved.

☞ **What is Isaac’s password?**

Slapper

After revealing Isaac’s username and password, you will also need to look into another user who used HTTP Basic Authorization to log in to their account.

Part 2- What is HTTP Basic Authentication?

HTTP Basic authentication tries to hide passwords with a very old, simple **encoding** (changing information into a secret code) process before sending them. This is not much better than sending them in cleartext (not encrypted, the way information was sent in Isaac’s case) because Wireshark automatically decodes (changing a secret code into a readable language) any information that is encoded through this process.

Even though this next user used a form of encoding called **HTTP Basic Authentication,** it is your job to try and discover the username and password of this person.

To do this, follow the steps below:

1. Download this file and double-click it to open it in Wireshark: BasicLogin.pcapng.
2. You are looking for this user’s credentials. In order to find his credentials, filter the packets with the “http” filter, and review each packet that is a GET request.
3. Inside the GET request (middle pane) You will need to locate the user’s “authorization,” which would be located inside the Hypertext Transfer Protocol dropdown menu.

Once you locate the credentials of this user, ask your classmates if they found the same information that you found.

Part 3 - Incident Reports

Now that you have identified the issues that occurred, you will need to make an incident report for each of them that will be shared with Sam.

Since there were two security issues, you will need to write two security reports. Your incident reports will include:

- Date and time of the activity.
- The source and destination IP address.
- The issue or security concern.
- Solutions or recommendations to correct the security issue.

You may make a copy of this template to create both of your incident reports.

Requirements

Upload your incident reports.

Reported By:	Date of Report:
Ibrana Choudhry	**JUne,23,24**
Signature:______________________________	

Incident Details

The user named Isaac attempted to log in to the system using the HTTP protocol. During this process, Isaac tried three different passwords before successfully logging in on his third attempt. Upon inspection using Wireshark, it was discovered that all three passwords were transmitted in plaintext and could be easily intercepted and read. HTTP protocol was used without any encryption, leading to potential data interception and security breach.

Another user tried to login using HTTP Basic Authorization to log in to his account. This method tries to hide passwords with a very old, simple encoding process but this is not much better than sending them in cleartext because Wireshark automatically decodes any information that is encoded through this process.

Date of Activity **June 28,2018**	**Time of Activity** **18:57**

Source IP Address **172.16.1.129**	**Destination IP Address** **172.16.1.1**
Source Mac Address **00:50:56:c0:00:08**	**Destination Mac Address** **00:0c:29:el:fe:3f**

Potential Solutions or Recommendations:

. ***Implement HTTPS***:

- Transition from HTTP to HTTPS to ensure that all data transmitted between the client and server is encrypted. This will prevent unauthorized interception and access to sensitive information.

Use Secure Authentication Methods:

- Implement multi-factor authentication (MFA) to add an additional layer of security.
- Ensure passwords are hashed and salted before storing or transmitting them.

Regular Security Audits:

- Conduct regular security audits and vulnerability assessments to identify and mitigate potential security risks.
- Use tools like Wireshark periodically to monitor for any unencrypted sensitive data transmission.

User Training and Awareness:

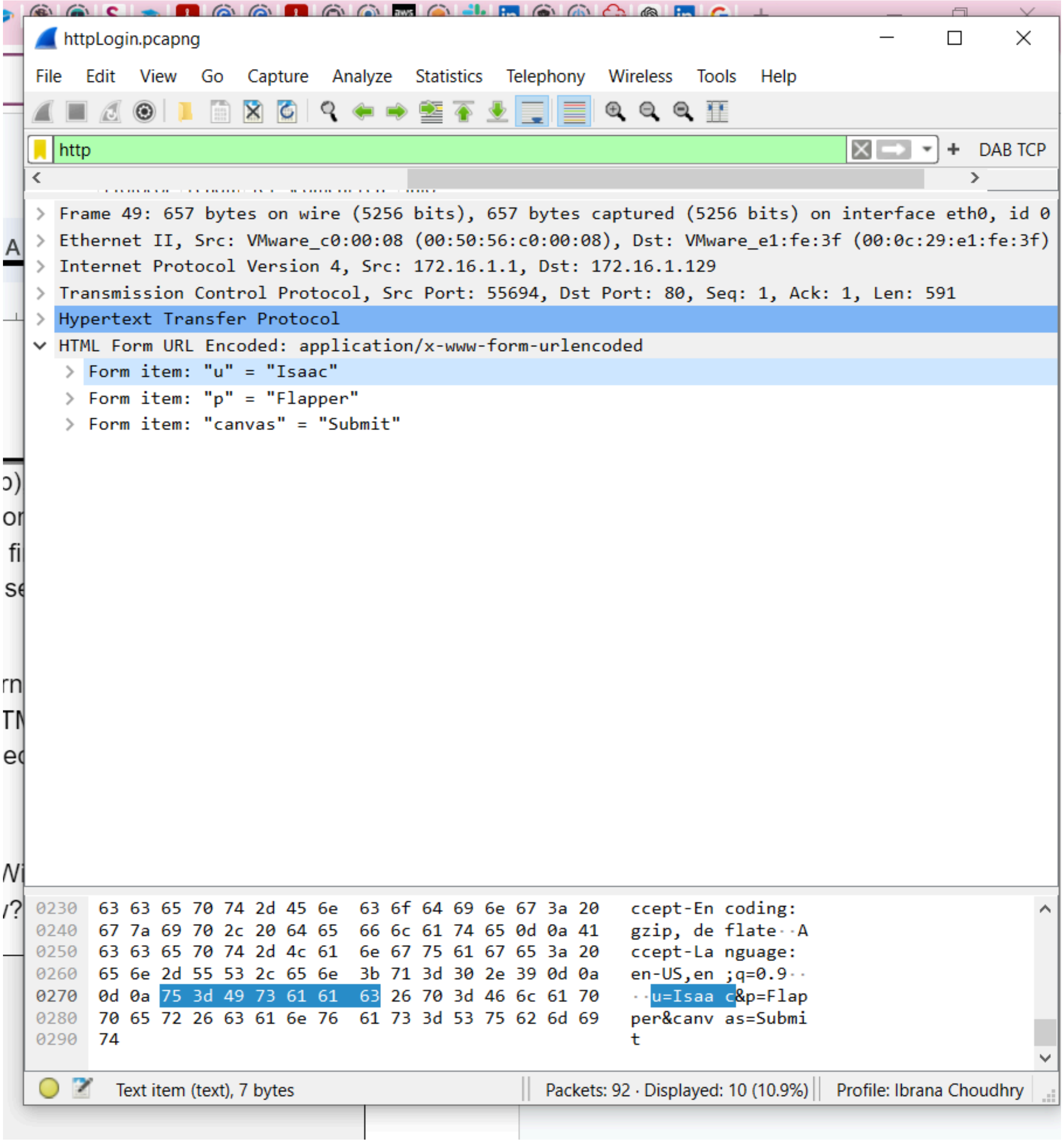
- Educate users about the importance of secure passwords and the risks associated with transmitting sensitive information over unsecured connections.
- Provide guidelines on recognizing and reporting suspicious activities.

Compliance with Standards:

- Ensure that the system complies with industry standards and regulations.

Incident Response Plan:

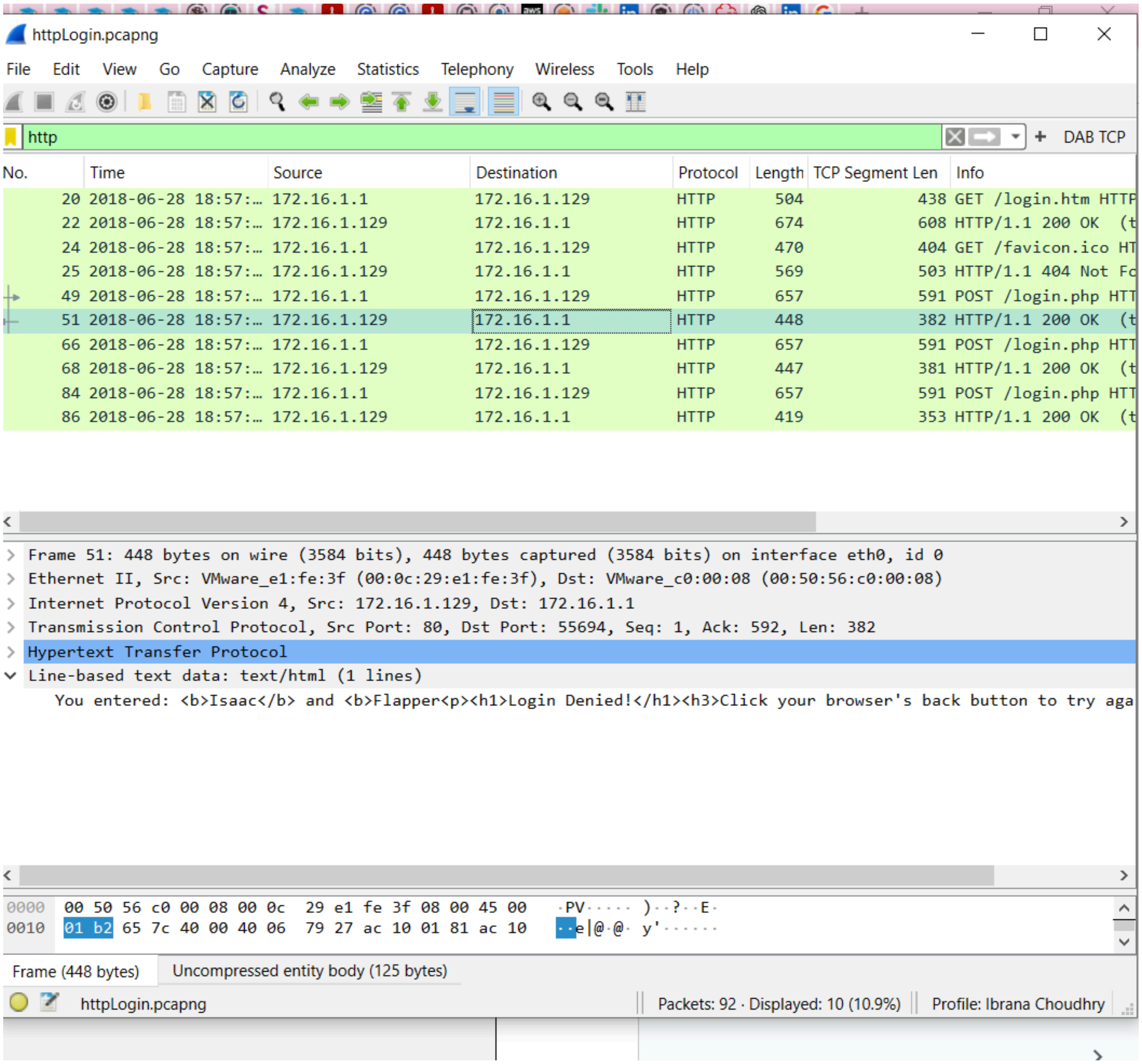
- Develop and implement a robust incident response plan to quickly address and mitigate the effects of security breaches.
- Ensure all team members are trained on the procedures to follow in case of a security incident.


httpLogin.pcapng
File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help
http
DAB TCP

```
> Frame 49: 657 bytes on wire (5256 bits), 657 bytes captured (5256 bits) on interface eth0, id 0
> Ethernet II, Src: VMware_c0:00:08 (00:50:56:c0:00:08), Dst: VMware_e1:fe:3f (00:0c:29:e1:fe:3f)
> Internet Protocol Version 4, Src: 172.16.1.1, Dst: 172.16.1.129
> Transmission Control Protocol, Src Port: 55694, Dst Port: 80, Seq: 1, Ack: 1, Len: 591
> Hypertext Transfer Protocol
v HTML Form URL Encoded: application/x-www-form-urlencoded
   > Form item: "u" = "Isaac"
   > Form item: "p" = "Flapper"
   > Form item: "canvas" = "Submit"
```

```
0230  63 63 65 70 74 2d 45 6e  63 6f 64 69 6e 67 3a 20   ccept-En coding: 
0240  67 7a 69 70 2c 20 64 65  66 6c 61 74 65 0d 0a 41   gzip, de flate··A
0250  63 63 65 70 74 2d 4c 61  6e 67 75 61 67 65 3a 20   ccept-La nguage: 
0260  65 6e 2d 55 53 2c 65 6e  3b 71 3d 30 2e 39 0d 0a   en-US,en ;q=0.9··
0270  0d 0a 75 3d 49 73 61 61  63 26 70 3d 46 6c 61 70   ··u=Isaa c&p=Flap
0280  70 65 72 26 63 61 6e 76  61 73 3d 53 75 62 6d 69   per&canv as=Submi
0290  74                                                 t
```

Text item (text), 7 bytes | Packets: 92 · Displayed: 10 (10.9%) | Profile: Ibrana Choudhry


httpLogin.pcapng
File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help
DAB TCP

http

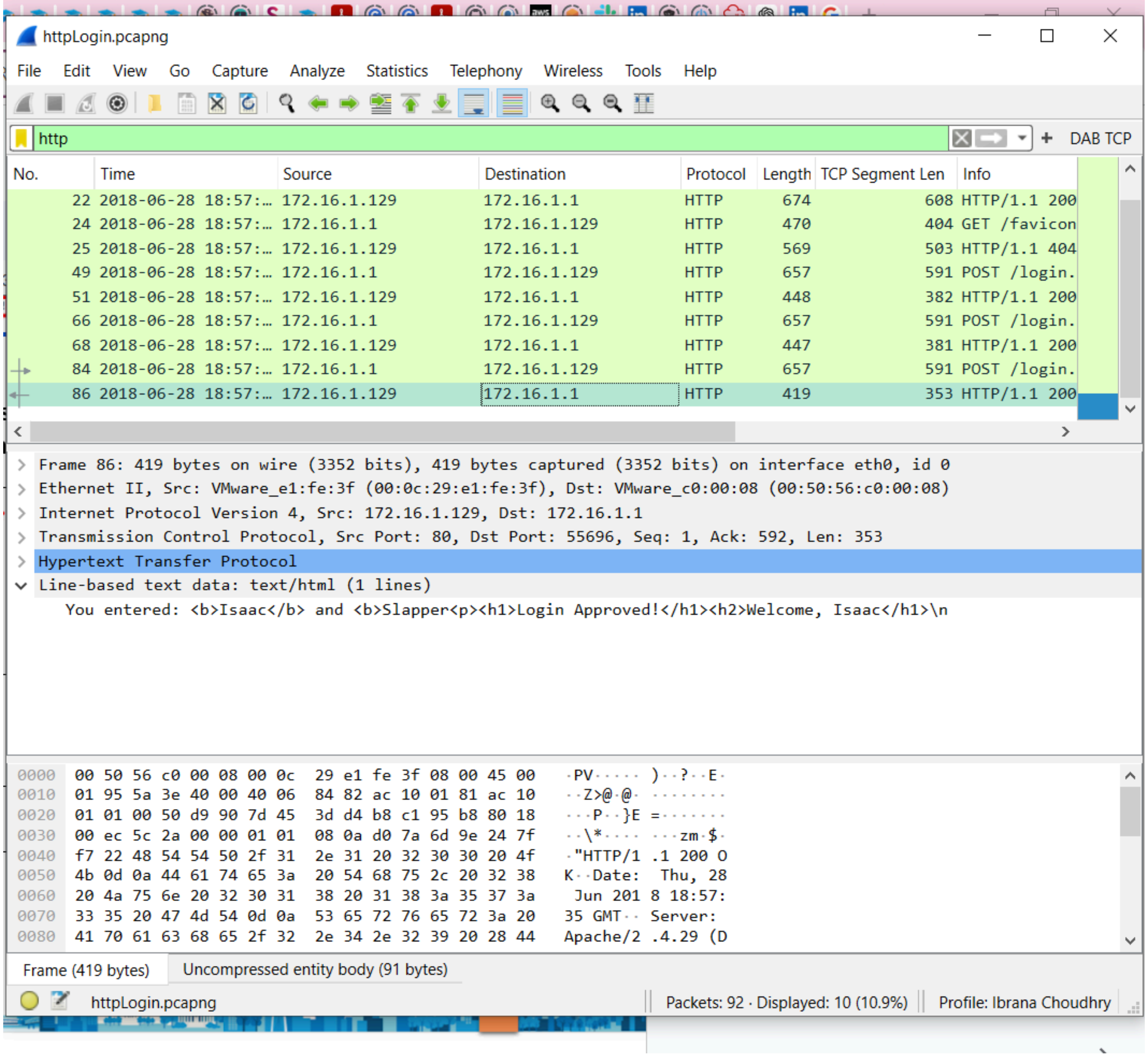
```
No.  Time                 Source        Destination   Protocol Length TCP Segment Len Info
  20 2018-06-28 18:57:… 172.16.1.1    172.16.1.129  HTTP     504         438 GET /login.htm HTTP
  22 2018-06-28 18:57:… 172.16.1.129  172.16.1.1    HTTP     674         608 HTTP/1.1 200 OK  (t
  24 2018-06-28 18:57:… 172.16.1.1    172.16.1.129  HTTP     470         404 GET /favicon.ico HT
  25 2018-06-28 18:57:… 172.16.1.129  172.16.1.1    HTTP     569         503 HTTP/1.1 404 Not Fo
  49 2018-06-28 18:57:… 172.16.1.1    172.16.1.129  HTTP     657         591 POST /login.php HTT
  51 2018-06-28 18:57:… 172.16.1.129  172.16.1.1    HTTP     448         382 HTTP/1.1 200 OK  (t
  66 2018-06-28 18:57:… 172.16.1.1    172.16.1.129  HTTP     657         591 POST /login.php HTT
  68 2018-06-28 18:57:… 172.16.1.129  172.16.1.1    HTTP     447         381 HTTP/1.1 200 OK  (t
  84 2018-06-28 18:57:… 172.16.1.1    172.16.1.129  HTTP     657         591 POST /login.php HTT
  86 2018-06-28 18:57:… 172.16.1.129  172.16.1.1    HTTP     419         353 HTTP/1.1 200 OK  (t
```

```
> Frame 51: 448 bytes on wire (3584 bits), 448 bytes captured (3584 bits) on interface eth0, id 0
> Ethernet II, Src: VMware_e1:fe:3f (00:0c:29:e1:fe:3f), Dst: VMware_c0:00:08 (00:50:56:c0:00:08)
> Internet Protocol Version 4, Src: 172.16.1.129, Dst: 172.16.1.1
> Transmission Control Protocol, Src Port: 80, Dst Port: 55694, Seq: 1, Ack: 592, Len: 382
> Hypertext Transfer Protocol
v Line-based text data: text/html (1 lines)
    You entered: <b>Isaac</b> and <b>Flapper<p><h1>Login Denied!</h1><h3>Click your browser's back button to try aga
```

```
0000  00 50 56 c0 00 08 00 0c  29 e1 fe 3f 08 00 45 00   ·PV·····  )··?··E·
0010  01 b2 65 7c 40 00 40 06  79 27 ac 10 01 81 ac 10   ··e|@·@·  y'······
```

Frame (448 bytes)   Uncompressed entity body (125 bytes)

httpLogin.pcapng   Packets: 92 · Displayed: 10 (10.9%)   Profile: Ibrana Choudhry



```
No.   Time                Source          Destination     Protocol  Length  TCP Segment Len  Info
   22 2018-06-28 18:57:… 172.16.1.129    172.16.1.1      HTTP      674              608 HTTP/1.1 200
   24 2018-06-28 18:57:… 172.16.1.1      172.16.1.129    HTTP      470              404 GET /favicon
   25 2018-06-28 18:57:… 172.16.1.129    172.16.1.1      HTTP      569              503 HTTP/1.1 404
   49 2018-06-28 18:57:… 172.16.1.1      172.16.1.129    HTTP      657              591 POST /login.
   51 2018-06-28 18:57:… 172.16.1.129    172.16.1.1      HTTP      448              382 HTTP/1.1 200
   66 2018-06-28 18:57:… 172.16.1.1      172.16.1.129    HTTP      657              591 POST /login.
   68 2018-06-28 18:57:… 172.16.1.129    172.16.1.1      HTTP      447              381 HTTP/1.1 200
   84 2018-06-28 18:57:… 172.16.1.1      172.16.1.129    HTTP      657              591 POST /login.
   86 2018-06-28 18:57:… 172.16.1.129    172.16.1.1      HTTP      419              353 HTTP/1.1 200
```

```
> Frame 86: 419 bytes on wire (3352 bits), 419 bytes captured (3352 bits) on interface eth0, id 0
> Ethernet II, Src: VMware_e1:fe:3f (00:0c:29:e1:fe:3f), Dst: VMware_c0:00:08 (00:50:56:c0:00:08)
> Internet Protocol Version 4, Src: 172.16.1.129, Dst: 172.16.1.1
> Transmission Control Protocol, Src Port: 80, Dst Port: 55696, Seq: 1, Ack: 592, Len: 353
> Hypertext Transfer Protocol
v Line-based text data: text/html (1 lines)
    You entered: <b>Isaac</b> and <b>Slapper<p><h1>Login Approved!</h1><h2>Welcome, Isaac</h1>\n
```

```
0000  00 50 56 c0 00 08 00 0c  29 e1 fe 3f 08 00 45 00   ·PV····· )··?··E·
0010  01 95 5a 3e 40 00 40 06  84 82 ac 10 01 81 ac 10   ··Z>@·@· ········
0020  01 01 00 50 d9 90 7d 45  3d d4 b8 c1 95 b8 80 18   ···P··}E =·······
0030  00 ec 5c 2a 00 00 01 01  08 0a d0 7a 6d 9e 24 7f   ··\*···· ···zm·$·
0040  f7 22 48 54 54 50 2f 31  2e 31 20 32 30 30 20 4f   ·"HTTP/1 .1 200 O
0050  4b 0d 0a 44 61 74 65 3a  20 54 68 75 2c 20 32 38   K··Date:  Thu, 28
0060  20 4a 75 6e 20 32 30 31  38 20 31 38 3a 35 37 3a    Jun 201 8 18:57:
0070  33 35 20 47 4d 54 0d 0a  53 65 72 76 65 72 3a 20   35 GMT·· Server: 
0080  41 70 61 63 68 65 2f 32  2e 34 2e 32 39 20 28 44   Apache/2 .4.29 (D
```

Frame (419 bytes)
Uncompressed entity body (91 bytes)
httpLogin.pcapng
Packets: 92 · Displayed: 10 (10.9%)
Profile: Ibrana Choudhry