

Http Basic Auth [Let's Defend – Write-up]

We receive a log indicating a possible attack, can you gather information from the .pcap file?

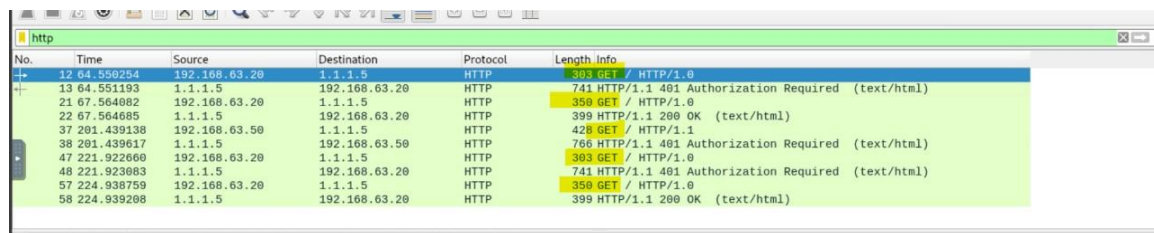
Log file: /root/Desktop/ChallengeFile/webserver.em0.pcap

Note: pcap file found public resources.

| Start Investigation

Q1: How many HTTP GET requests are in pcap?

After I opened the pcap file, I searched for http to get the GET method.



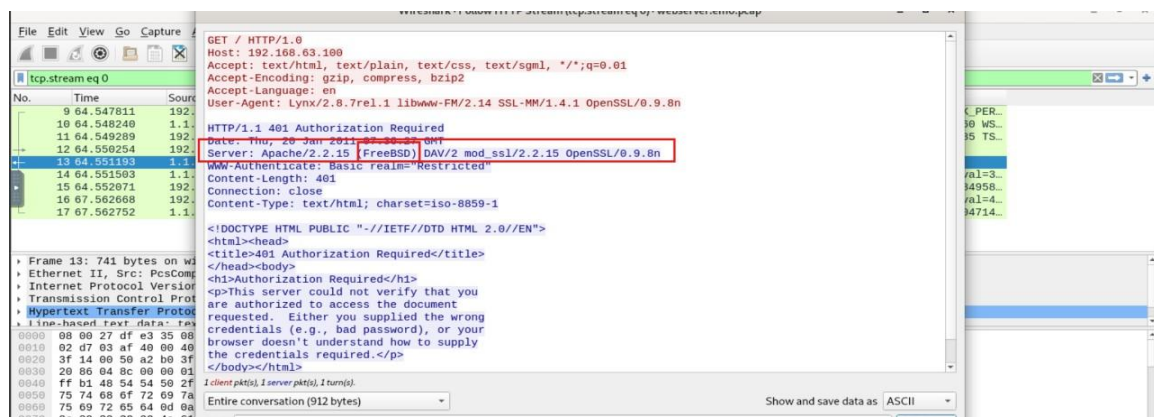
A screenshot of the Wireshark network protocol analyzer. The 'http' filter is applied in the display filter bar. The packet list pane shows several HTTP packets. The packet details pane for packet 13 (GET / HTTP/1.0) is expanded, showing the request line and headers.

No.	Time	Source	Destination	Protocol	Length	Info
12	64.550254	192.168.63.20	192.168.63.20	HTTP	350	GET / HTTP/1.0
13	64.551193	1.1.1.5	192.168.63.20	HTTP	741	HTTP/1.1 401 Authorization Required (text/html)
21	67.564082	192.168.63.20	1.1.1.5	HTTP	350	GET / HTTP/1.0
22	67.564085	1.1.1.5	192.168.63.20	HTTP	399	HTTP/1.1 200 OK (text/html)
37	201.439138	192.168.63.50	1.1.1.5	HTTP	428	GET / HTTP/1.1
38	201.439617	1.1.1.5	192.168.63.50	HTTP	766	HTTP/1.1 401 Authorization Required (text/html)
47	221.922600	192.168.63.20	1.1.1.5	HTTP	303	GET / HTTP/1.0
48	221.923083	1.1.1.5	192.168.63.20	HTTP	741	HTTP/1.1 401 Authorization Required (text/html)
57	224.938759	192.168.63.20	1.1.1.5	HTTP	350	GET / HTTP/1.0
58	224.939208	1.1.1.5	192.168.63.20	HTTP	399	HTTP/1.1 200 OK (text/html)

Answer: 5

Q2: What is the server operating system?

After that, I checked one of the packets in question 1 and then searched for a server.



A screenshot of the Wireshark network protocol analyzer. The 'tcp.stream eq 0' filter is applied. The packet list pane shows several packets. The packet details pane for packet 13 (HTTP/1.1 401 Authorization Required) is expanded, showing the response line and headers. The packet bytes pane shows the raw data of the packet.

No.	Time	Source	Destination	Protocol	Length	Info
9	64.547811	192.168.63.20	192.168.63.20	HTTP	350	GET / HTTP/1.0
10	64.548240	1.1.1.5	192.168.63.20	HTTP	741	HTTP/1.1 401 Authorization Required
11	64.548289	192.168.63.20	1.1.1.5	HTTP	350	GET / HTTP/1.0
12	64.550254	192.168.63.20	1.1.1.5	HTTP	350	GET / HTTP/1.0
13	64.551193	1.1.1.5	192.168.63.20	HTTP	741	HTTP/1.1 401 Authorization Required
14	64.551503	1.1.1.5	192.168.63.20	HTTP	399	HTTP/1.1 200 OK (text/html)
15	64.552071	192.168.63.20	1.1.1.5	HTTP	350	GET / HTTP/1.0
16	67.562608	1.1.1.5	192.168.63.20	HTTP	766	HTTP/1.1 401 Authorization Required
17	67.562752	1.1.1.5	192.168.63.20	HTTP	399	HTTP/1.1 200 OK (text/html)

Answer: freebsd

Q3: What is the name and version of the web server software?

Also, the answer in the same packet

```
HTTP/1.1 401 Authorization Required
Date: Thu, 20 Jan 2011 07:36:27 GMT
Server: Apache/2.2.15 (FreeBSD) DAV/2 mod_ssl/2.2.15 OpenSSL/0.9.8n
WWW-Authenticate: Basic realm="Restricted"
Content-Length: 401
Connection: close
Content-Type: text/html; charset=iso-8859-1

<!DOCTYPE HTML PUBLIC "-//IETF//DTD HTML 2.0//EN">
```

Answer: apache/2.2.15

Q4: What is the version of OpenSSL running on the server?

Also, the answer in the same packet

```
HTTP/1.1 401 Authorization Required
Date: Thu, 20 Jan 2011 07:36:27 GMT
Server: Apache/2.2.15 (FreeBSD) DAV/2 mod_ssl/2.2.15 OpenSSL/0.9.8n
WWW-Authenticate: Basic realm="Restricted"
Content-Length: 401
Connection: close
Content-Type: text/html; charset=iso-8859-1

<!DOCTYPE HTML PUBLIC "-//IETF//DTD HTML 2.0//EN">
<html><head>
<title>401 Authorization Required</title>
</head><body>
<h1>Authorization Required</h1>
<p>This server could not verify that you
are authorized to access the document
```

Answer: openssl/0.9.8n

Q5: What is the client's user-agent information?

Also, the answer in the same packet

```
GET / HTTP/1.0
Host: 192.168.63.100
Accept: text/html, text/plain, text/css, text/sgml, */*;q=0.01
Accept-Encoding: gzip, compress, bzip2
Accept-Language: en
User-Agent: Lynx/2.8.7rel.1 libwww-FM/2.14 SSL-MM/1.4.1 OpenSSL/0.9.8n

HTTP/1.1 401 Authorization Required
```

Answer: lynx/2.8.7rel.1 libwww-fm/2.14 ssl-mm/1.4.1 openssl/0.9.8n

Q6: What is the username used for Basic Authentication?

I checked one of the packets and got the username and password for base64

```
GET / HTTP/1.0
Host: 192.168.63.100
Accept: text/html, text/plain, text/css, text/sgml, */*;q=0.01
Accept-Encoding: gzip, compress, bzip2
Accept-Language: en
User-Agent: Lynx/2.8.7rel.1 libwww-FM/2.14 SSL-MM/1.4.1 OpenSSL/0.9.8n
Authorization: Basic d2ViYWRTaW46VzNiNERtMW4=

HTTP/1.1 200 OK
Date: Thu, 20 Jan 2011 07:39:08 GMT
Server: Apache/2.2.15 (FreeBSD) DAV/2 mod_ssl/2.2.15 OpenSSL/0.9.8n
Last-Modified: Mon, 27 Dec 2010 13:11:28 GMT
ETag: "3f44b-2c-4986412e52000"
Accept-Ranges: bytes
Content-Length: 44
Connection: close
Content-Type: text/html

<html><body><h1>It works!</h1></body></html>
```

The screenshot shows the CyberChef web interface. On the left, the 'Operations' sidebar lists various tools. The main 'Recipe' area shows a 'From Base64' operation selected, with a dropdown menu set to 'Alphabet' and 'A-Za-z0-9+/=' selected. The 'Remove non-alphabet chars' checkbox is checked. The 'Input' field on the right contains the Base64 string 'd2ViYWRTaW46VzNiNERtMW4=' and the 'Output' field displays the decoded result 'webadmin:w3b4Dm1n'.

So I cracked it using cyberchef and got the username and password, this answer for questions 6 and 7.

Answer: webadmin

Q7: What is the user password used for Basic Authentication?

Answer: w3b4dm1n

Summary

This investigation into HTTP Basic Authentication was an eye-opening experience that significantly improved my network forensics and packet analysis skills.

Through this lab, I learned how to move beyond basic traffic monitoring to performing deep packet inspection (DPI) using Wireshark. It was fascinating to see how a server's entire "fingerprint"—including its OS (FreeBSD), web server version (Apache/2.2.15), and even the OpenSSL version—is clearly visible within HTTP response headers.

The most valuable lesson, however, was the hands-on demonstration of why Basic Authentication is a major security risk. Seeing the Authorization header in the packet and realizing it was just a simple Base64 string that I could easily decode with CyberChef to reveal the username (webadmin) and password (w3b4dm1n) made the concept of "clear-text credentials" very real to me. This investigation perfectly bridged the gap between theoretical security risks and actual, exploitable vulnerabilities.