Strange HTTP Headers

Related PCAP File(s):

- `CRLF_and_host_header_manipulation.pcapng`

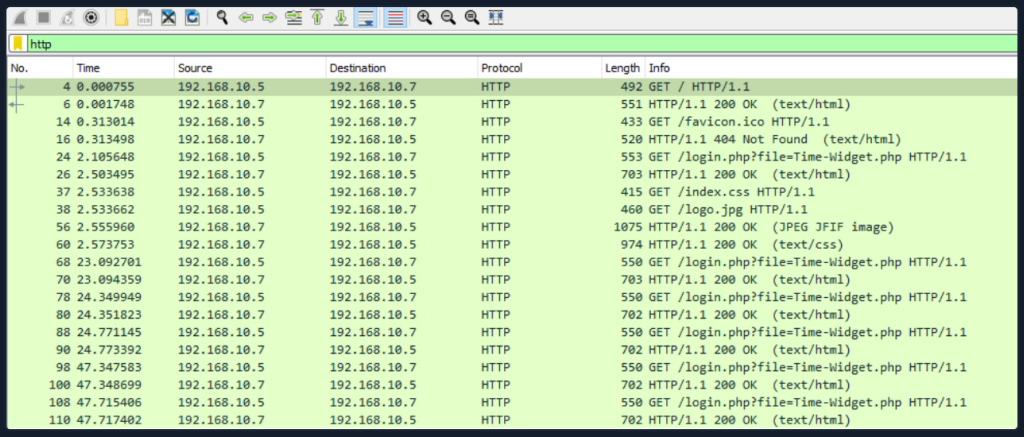
We might not notice anything like fuzzing right away when analyzing our web server's traffic. However, this does not always indicate that nothing bad is happening. Instead, we can always look a little bit deeper. In order to do so, we might look for strange behavior among HTTP requests. Some of which are weird headers like

1. `Weird Hosts (Host: )`
2. `Unusual HTTP Verbs`
3. `Changed User Agents`

Finding Strange Host Headers

In order to start, as we would normally do, we can limit our view in Wireshark to only http replies and requests.

- `http`

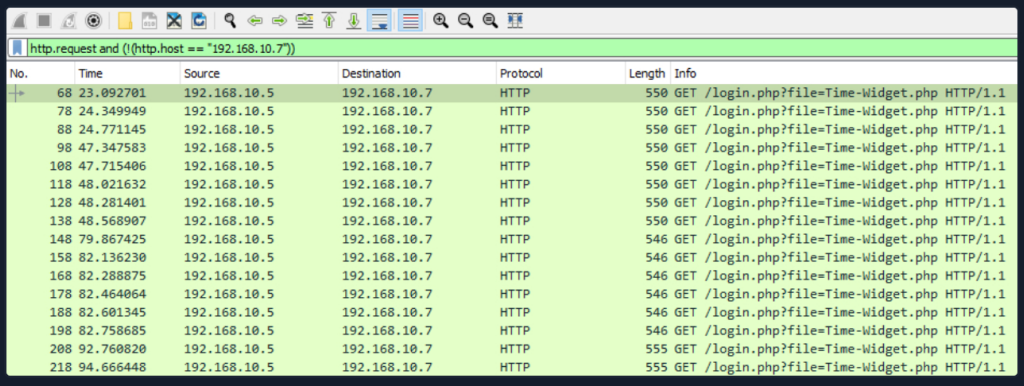


http

No.	Time	Source	Destination	Protocol	Length	Info
4	0.000755	192.168.10.5	192.168.10.7	HTTP	492	GET / HTTP/1.1
6	0.001748	192.168.10.7	192.168.10.5	HTTP	551	HTTP/1.1 200 OK (text/html)
14	0.313014	192.168.10.5	192.168.10.7	HTTP	433	GET /favicon.ico HTTP/1.1
16	0.313498	192.168.10.7	192.168.10.5	HTTP	520	HTTP/1.1 404 Not Found (text/html)
24	2.105648	192.168.10.5	192.168.10.7	HTTP	553	GET /login.php?file=Time-Widget.php HTTP/1.1
26	2.503495	192.168.10.7	192.168.10.5	HTTP	703	HTTP/1.1 200 OK (text/html)
37	2.533638	192.168.10.5	192.168.10.7	HTTP	415	GET /index.css HTTP/1.1
38	2.533662	192.168.10.5	192.168.10.7	HTTP	460	GET /logo.jpg HTTP/1.1
56	2.555960	192.168.10.7	192.168.10.5	HTTP	1075	HTTP/1.1 200 OK (JPEG JFIF image)
60	2.573753	192.168.10.7	192.168.10.5	HTTP	974	HTTP/1.1 200 OK (text/css)
68	23.092701	192.168.10.5	192.168.10.7	HTTP	550	GET /login.php?file=Time-Widget.php HTTP/1.1
70	23.094359	192.168.10.7	192.168.10.5	HTTP	703	HTTP/1.1 200 OK (text/html)
78	24.349949	192.168.10.5	192.168.10.7	HTTP	550	GET /login.php?file=Time-Widget.php HTTP/1.1
80	24.351823	192.168.10.7	192.168.10.5	HTTP	702	HTTP/1.1 200 OK (text/html)
88	24.771145	192.168.10.5	192.168.10.7	HTTP	550	GET /login.php?file=Time-Widget.php HTTP/1.1
90	24.773392	192.168.10.7	192.168.10.5	HTTP	702	HTTP/1.1 200 OK (text/html)
98	47.347583	192.168.10.5	192.168.10.7	HTTP	550	GET /login.php?file=Time-Widget.php HTTP/1.1
100	47.348699	192.168.10.7	192.168.10.5	HTTP	702	HTTP/1.1 200 OK (text/html)
108	47.715406	192.168.10.5	192.168.10.7	HTTP	550	GET /login.php?file=Time-Widget.php HTTP/1.1
110	47.717402	192.168.10.7	192.168.10.5	HTTP	702	HTTP/1.1 200 OK (text/html)

Then, we can find any irregular Host headers with the following command. We specify our web server's real IP address to exclude any entries which use this real header. If we were to do this for an external web server, we could specify the domain name here.

- `http.request and (!(http.host == "192.168.10.7"))`



http.request and (!(http.host == "192.168.10.7"))

No.	Time	Source	Destination	Protocol	Length	Info
68	23.092701	192.168.10.5	192.168.10.7	HTTP	550	GET /login.php?file=Time-Widget.php HTTP/1.1
78	24.349949	192.168.10.5	192.168.10.7	HTTP	550	GET /login.php?file=Time-Widget.php HTTP/1.1
88	24.771145	192.168.10.5	192.168.10.7	HTTP	550	GET /login.php?file=Time-Widget.php HTTP/1.1
98	47.347583	192.168.10.5	192.168.10.7	HTTP	550	GET /login.php?file=Time-Widget.php HTTP/1.1
108	47.715406	192.168.10.5	192.168.10.7	HTTP	550	GET /login.php?file=Time-Widget.php HTTP/1.1
118	48.021632	192.168.10.5	192.168.10.7	HTTP	550	GET /login.php?file=Time-Widget.php HTTP/1.1
128	48.281401	192.168.10.5	192.168.10.7	HTTP	550	GET /login.php?file=Time-Widget.php HTTP/1.1
138	48.568907	192.168.10.5	192.168.10.7	HTTP	550	GET /login.php?file=Time-Widget.php HTTP/1.1
148	79.867425	192.168.10.5	192.168.10.7	HTTP	546	GET /login.php?file=Time-Widget.php HTTP/1.1
158	82.136230	192.168.10.5	192.168.10.7	HTTP	546	GET /login.php?file=Time-Widget.php HTTP/1.1
168	82.288875	192.168.10.5	192.168.10.7	HTTP	546	GET /login.php?file=Time-Widget.php HTTP/1.1
178	82.464064	192.168.10.5	192.168.10.7	HTTP	546	GET /login.php?file=Time-Widget.php HTTP/1.1
188	82.601345	192.168.10.5	192.168.10.7	HTTP	546	GET /login.php?file=Time-Widget.php HTTP/1.1
198	82.758685	192.168.10.5	192.168.10.7	HTTP	546	GET /login.php?file=Time-Widget.php HTTP/1.1
208	92.760820	192.168.10.5	192.168.10.7	HTTP	555	GET /login.php?file=Time-Widget.php HTTP/1.1
218	94.666448	192.168.10.5	192.168.10.7	HTTP	555	GET /login.php?file=Time-Widget.php HTTP/1.1

Suppose we noticed that this filter returned some results, we could dig into these HTTP requests a little deeper to find out what hosts these bad actors might have tried to use. We might commonly notice `127.0.0.1`.

```
> Frame 68: 550 bytes on wire (4400 bits), 550 bytes captured (4400 bits) on interface \Device\NPF_{CCC4B960-1E92-4BD5-BBF3-11E2DFD12FE1}, id 0
> Ethernet II, Src: PcsCompu_53:0c:ba (08:00:27:53:0c:ba), Dst: Micro-St_95:68:2a (44:8a:5b:95:68:2a)
> Internet Protocol Version 4, Src: 192.168.10.5, Dst: 192.168.10.7
> Transmission Control Protocol, Src Port: 32916, Dst Port: 80, Seq: 1, Ack: 1, Len: 484
v Hypertext Transfer Protocol
  > GET /login.php?file=Time-Widget.php HTTP/1.1\r\n
    Host: 127.0.0.1\r\n
    Upgrade-Insecure-Requests: 1\r\n
    User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/113.0.5672.93 Safari/537.36\r\n
    Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0…
    Referer: http://192.168.10.7/\r\n
    Accept-Encoding: gzip, deflate\r\n
    Accept-Language: en-US,en;q=0.9\r\n
    Connection: close\r\n
    \r\n
    [Full request URI: http://127.0.0.1/login.php?file=Time-Widget.php]
    [HTTP request 1/1]
    [Response in frame: 70]
```

Or instead something like admin.

```
> Frame 148: 546 bytes on wire (4368 bits), 546 bytes captured (4368 bits) on interface \Device\NPF_{CCC4B960-1E92-4BD5-BBF3-11E2DFD12FE1}, id 0
> Ethernet II, Src: PcsCompu_53:0c:ba (08:00:27:53:0c:ba), Dst: Micro-St_95:68:2a (44:8a:5b:95:68:2a)
> Internet Protocol Version 4, Src: 192.168.10.5, Dst: 192.168.10.7
> Transmission Control Protocol, Src Port: 36022, Dst Port: 80, Seq: 1, Ack: 1, Len: 480
v Hypertext Transfer Protocol
  > GET /login.php?file=Time-Widget.php HTTP/1.1\r\n
    Host: admin\r\n
    Upgrade-Insecure-Requests: 1\r\n
    User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/113.0.5672.93 Safari/537.36\r\n
    Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.7
    Referer: http://192.168.10.7/\r\n
    Accept-Encoding: gzip, deflate\r\n
    Accept-Language: en-US,en;q=0.9\r\n
    Connection: close\r\n
    \r\n
    [Full request URI: http://admin/login.php?file=Time-Widget.php]
    [HTTP request 1/1]
    [Response in frame: 150]
```

Attackers will attempt to use different host headers to gain levels of access they would not normally achieve through the legitimate host. They may use proxy tools like burp suite or others to modify these before sending them to the server. In order to prevent successful exploitation beyond only detecting these events, we should always do the following.

1. `Ensure that our virtualhosts or access configurations are setup correctly to prevent this form of access.`
2. `Ensure that our web server is up to date.`

Analyzing Code 400s and Request Smuggling

We might also notice some bad responses from our web server, like code 400s. These codes indicate a bad request from the client, so they can be a good place to start when detecting malicious actions via http/https. In order to filter for these, we can use the following

- `http.response.code == 400`

http.response.code == 400

No.	Time	Source	Destination	Protocol	Length	Info
230	219.249840	192.168.10.7	192.168.10.5	HTTP	549	HTTP/1.1 400 Bad Request (text/html)
240	220.925435	192.168.10.7	192.168.10.5	HTTP	549	HTTP/1.1 400 Bad Request (text/html)
280	373.130240	192.168.10.7	192.168.10.5	HTTP	549	HTTP/1.1 400 Bad Request (text/html)
290	375.462469	192.168.10.7	192.168.10.5	HTTP	549	HTTP/1.1 400 Bad Request (text/html)
300	375.949831	192.168.10.7	192.168.10.5	HTTP	549	HTTP/1.1 400 Bad Request (text/html)
310	376.286971	192.168.10.7	192.168.10.5	HTTP	549	HTTP/1.1 400 Bad Request (text/html)
320	376.594220	192.168.10.7	192.168.10.5	HTTP	549	HTTP/1.1 400 Bad Request (text/html)

Suppose we were to follow one of these HTTP streams, we might notice the following from the client.

```
GET /login.php?id=1%20HTTP/1.1%0d%0aHost:%20192.168.10.5%0d%0a%0d%0aGET%20/uploads/cmd.php%20HTTP/1.1%0d%0aHost:%20127.0.0.1:8080%0d%0a%0d%0a HTTP/1.13a8080%0d%0a%0d%0a%20 HTTP/1.1
Host: 192.168.10.5
Upgrade-Insecure-Requests: 1
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/113.0.5672.93 Safari/537.36
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.7
Referer: http://192.168.10.7/
```

This is commonly referred to as HTTP request smuggling or CRLF (Carriage Return Line Feed). Essentially, an attacker will try the following.

- `GET%20%2flogin.php%3fid%3d1%20HTTP%2f1.1%0d%0aHost%3a%20192.168.10.5%0d%0a%0d%0aGET%20%2fuploads%2fcmd2.php%20HTTP%2f1.1%0d%0aHost%3a%20127.0.0.1%3a8080%0d%0a%0d%0a%20HTTP%2f1.1 Host: 192.168.10.5`

Which will be decoded by our server like this.

Code: decoded

```
GET /login.php?id=1 HTTP/1.1
```

```
Host: 192.168.10.5

GET /uploads/cmd2.php HTTP/1.1
Host: 127.0.0.1:8080

 HTTP/1.1
Host: 192.168.10.5
```

Essentially, in cases where our configurations are vulnerable, the first request will go through, and the second request will as well shortly after. This can give an attacker levels of access that we would normally prohibit. This occurs due to our configuration looking like the following.

Apache Configuration

Code: txt

```
<VirtualHost *:80>

    RewriteEngine on
    RewriteRule "^/categories/(.*)" "http://192.168.10.100:8080/categories.php?id=$1" [P]
    ProxyPassReverse "/categories/" "http://192.168.10.100:8080/"

</VirtualHost>
```

CVE-2023-25690

As such watching for these code 400s can give clear indication to adversarial actions during our traffic analysis efforts. Additionally, we would notice if an attacker is successful with this attack by finding the code `200` (`success`) in response to one of the requests which look like this.

Waiting to start...

Enable step-by-step solutions for all questions

Questions

Answer the question(s) below to complete this Section and earn cubes!

+ 1 Inspect the CRLF_and_host_header_manipulation.pcapng file, part of this module's resources, and enter the total number of HTTP packets with response code 400 as your answer.

7

Submit

Previous Next

Mark Complete & Next