

HackerFrogs Afterschool

OverTheWire Natas: Part 2

Class:

Web App Hacking

Workshop Number:

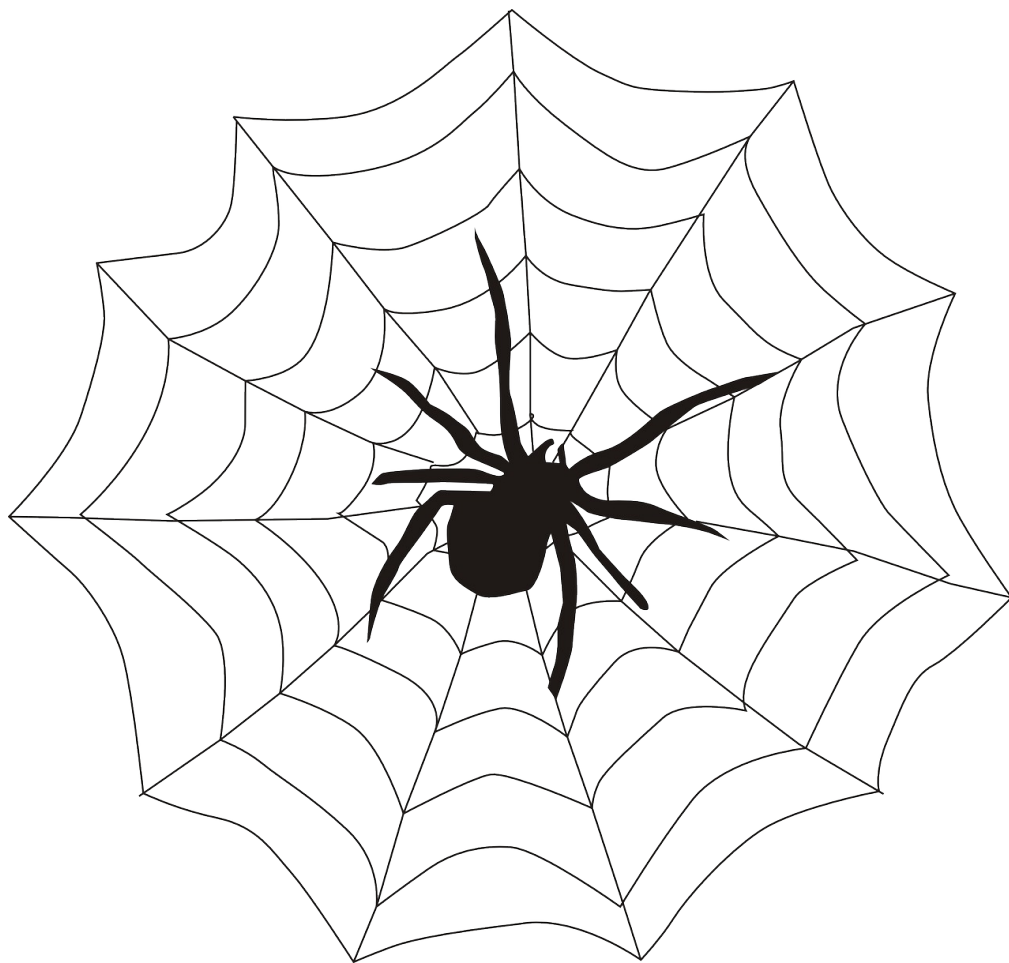
AS-WEB-02

Document Version:

1.75

Special Requirements:

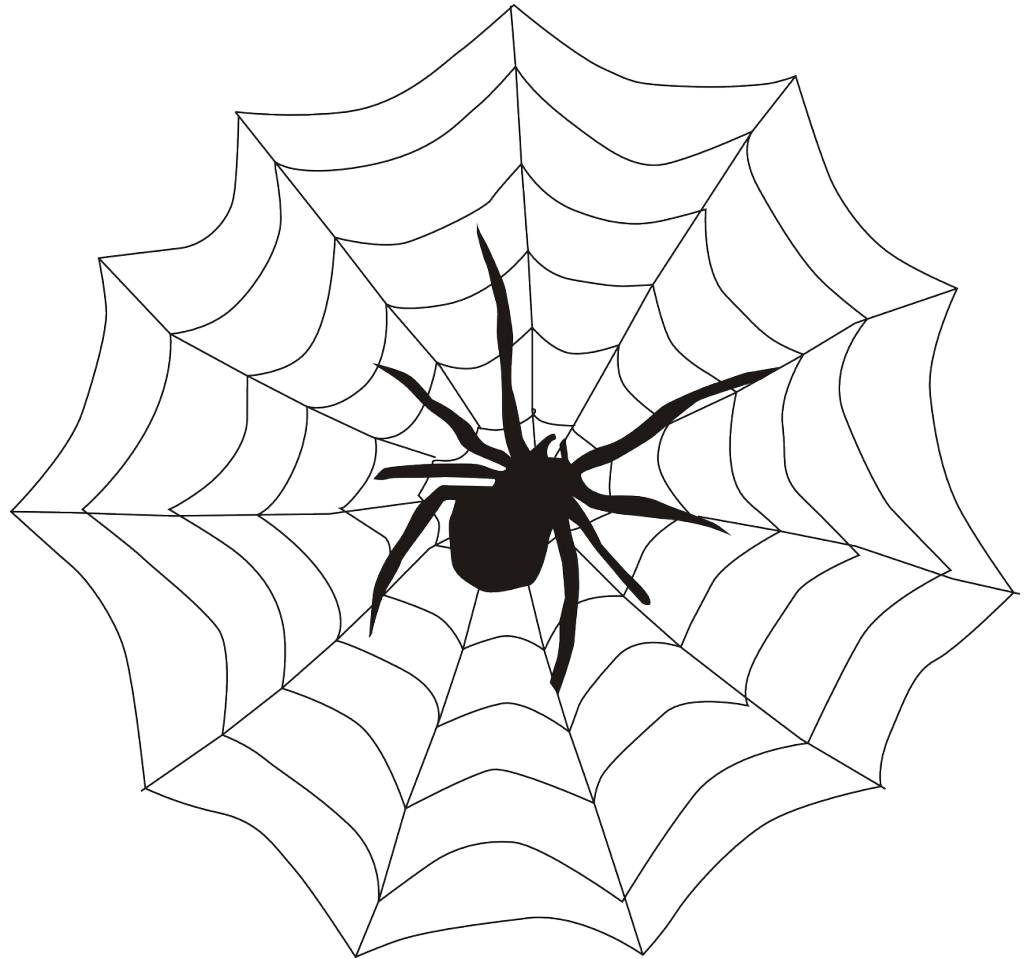
None



Web App Hacking

This is the second workshop in our intro to web app hacking course.

Let's take a few moments to review the concepts we learned in the previous workshop.



Directory Indexing

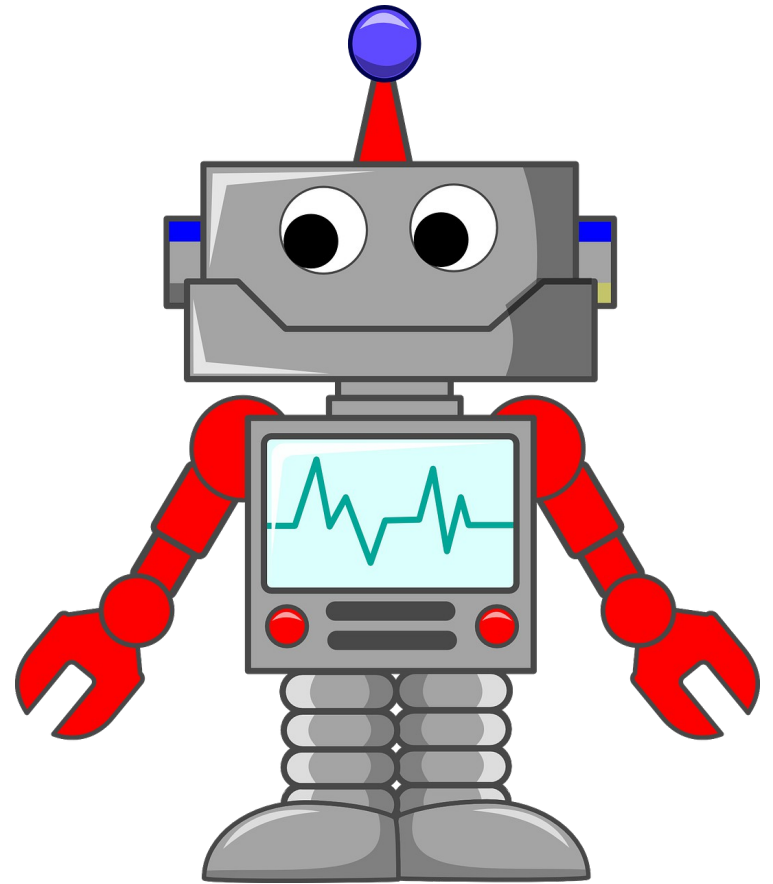
Directory indexing is a vulnerability on websites that allows users to see the file contents of web directories, which can lead to sensitive data exposed to arbitrary website visitors.

Directory listing

- [admin.html](#)
- [passwords.txt](#)
- [user_database.bak](#)

Robots.txt

Robots.txt is a special file included on certain websites that indicate which files on the site can / cannot be indexed by search engines. It can be abused to point malicious users to sensitive parts of the website.



Natas CTF

Let's continue with with the Natas CTF game at the following URL:

<http://natas4.natas.labs.overthewire.org/>

P-4 HTTP Headers

Each time a web browser accesses a webpage, the browser makes an HTTP request to the server that hosts the page.



P-4 HTTP Headers

In each HTTP request, several headers and their values are passed along to the server to ensure that the browser and server can communicate properly.



P-4 HTTP Headers

Some examples of HTTP headers and what info they provide to the web server:

Host	← the website being contacted e.g., <code>natas4.overthewire.org</code>
User-Agent	← the type of browser that is making the request e.g., <code>Chrome/0.2</code>
Accept	← the type of data that should be sent in response e.g., <code>*/*</code> (any type of data)

P-4 HTTP Headers

Some examples of HTTP headers and what info they provide to the web server:

Host	← the website being contacted e.g., natas4.overthewire.org
User-Agent	← the type of browser that is making the request e.g., Chrome/0.2
Accept	← the type of data that should be sent in response e.g., */* (any type of data)

P-4 HTTP Headers

Some examples of HTTP headers and what info they provide to the web server:

Host	← the website being contacted e.g., <code>natas4.overthewire.org</code>
User-Agent	← the type of browser that is making the request e.g., <code>Chrome/0.2</code>
Accept	← the type of data that should be sent in response e.g., <code>/*/*</code> (any type of data)

P-4 HTTP Headers

Some examples of HTTP headers and what info they provide to the web server:

Host	← the website being contacted e.g., <code>natas4.overthewire.org</code>
User-Agent	← the type of browser that is making the request e.g., <code>Chrome/0.2</code>
Accept	← the type of data that should be sent in response e.g., <code>/*/*</code> (any type of data)

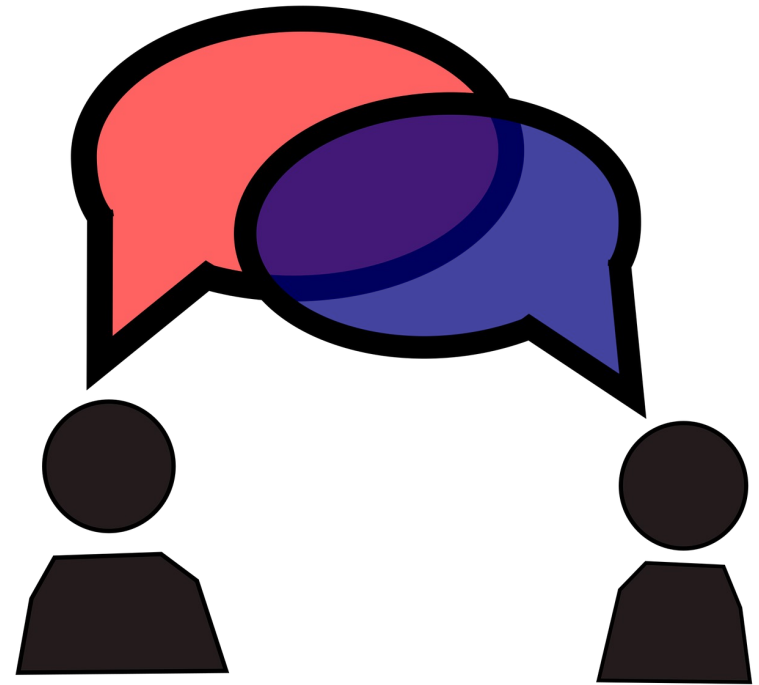
P-4 HTTP Headers

Please keep in mind that because HTTP headers can be modified by the user before being sent, that means that the values of any HTTP headers could be spoofed (falsified), although default web browser behavior doesn't allow this.



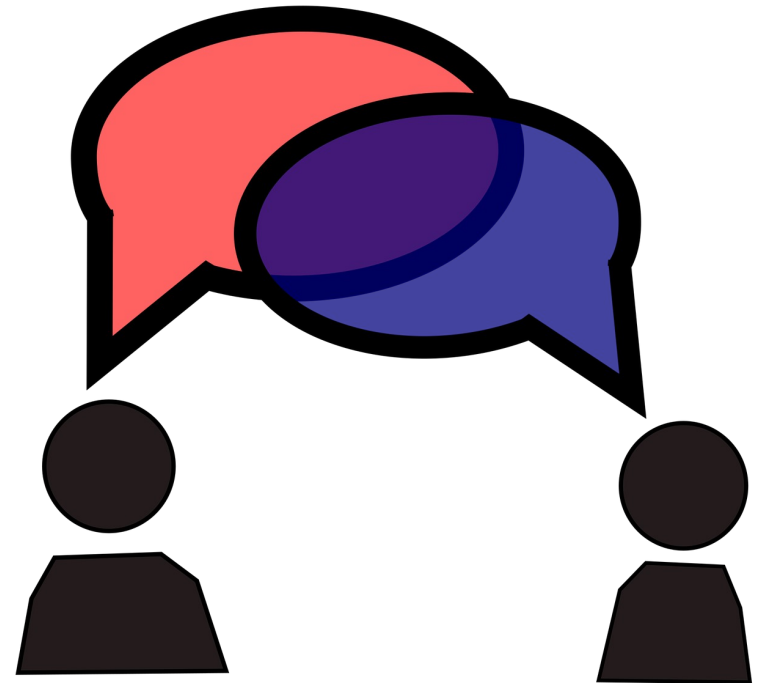
P-4 HTTP Referer Header

The HTTP Referer header (which is misspelled on purpose) contains the value of a complete or partial address of the webpage that is making the request.



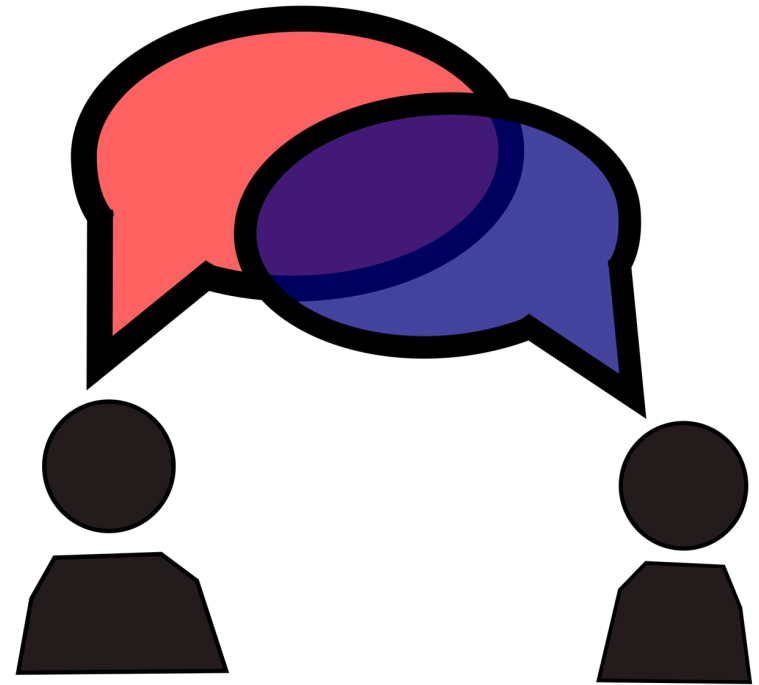
P-4 HTTP Referer Header

This allows the web server to identify which webpage users are visiting it from.



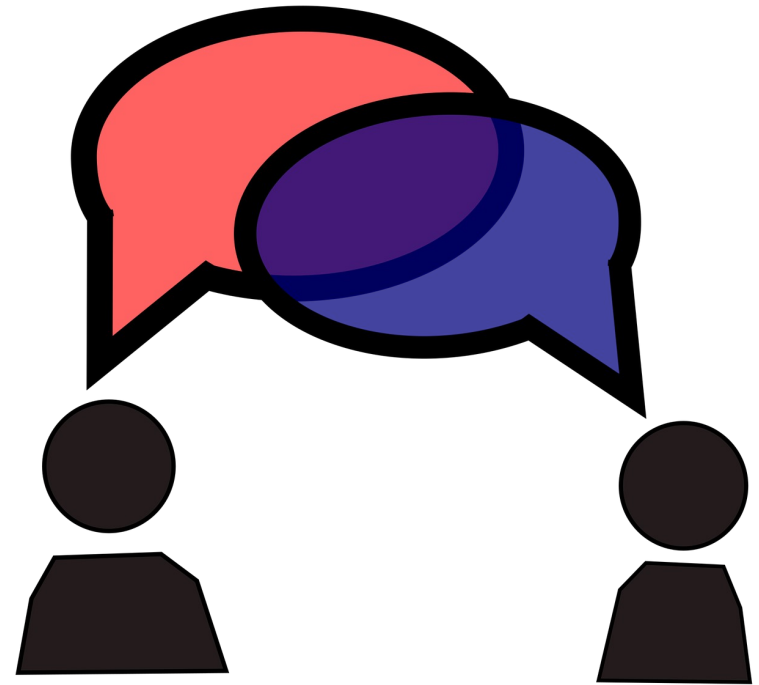
P-4 HTTP Referer Header

The data from this header can be useful for analytics and logging, etc.



P-4 HTTP Referer Header

However, some developers attempt to use the value of the Referer header as a type of security mechanism, for which it was not designed



The cURL Program

```
C:\Users\shyhat>curl -v -H Referer:http://natas5.natas.labs.overthewire.org/ -u
natas4:tK0cJIbzM4lTs8hbCmzn5Zr4434fGZQm http://natas4.natas.labs.overthewire.org

* Trying 13.50.142.37:80...
* Connected to natas4.natas.labs.overthewire.org (13.50.142.37) port 80 (#0)
* Server auth using Basic with user 'natas4'
> GET / HTTP/1.1
> Host: natas4.natas.labs.overthewire.org
> Authorization: Basic bmF0YXM0OnRLT2NKSWJ6TTRsVHM4aGJD bXpuNVpyNDQzNGZHWlFt
```

The cURL program is a command line interface (CLI) app that is common to all major computer operating systems.

The cURL Program

```
C:\Users\shyhat>curl -v -H Referer:http://natas5.natas.labs.overthewire.org/ -u
natas4:tK0cJIbzM4lTs8hbCmzn5Zr4434fGZQm http://natas4.natas.labs.overthewire.org

* Trying 13.50.142.37:80...
* Connected to natas4.natas.labs.overthewire.org (13.50.142.37) port 80 (#0)
* Server auth using Basic with user 'natas4'
> GET / HTTP/1.1
> Host: natas4.natas.labs.overthewire.org
> Authorization: Basic bmF0YXM0OnRLT2NKSWJ6TTRsVHM4aGJDbXpuNVpyNDQzNGZHWlFt
```

The program allows for access to webpages from the CLI, but only returns text, such as HTML code.

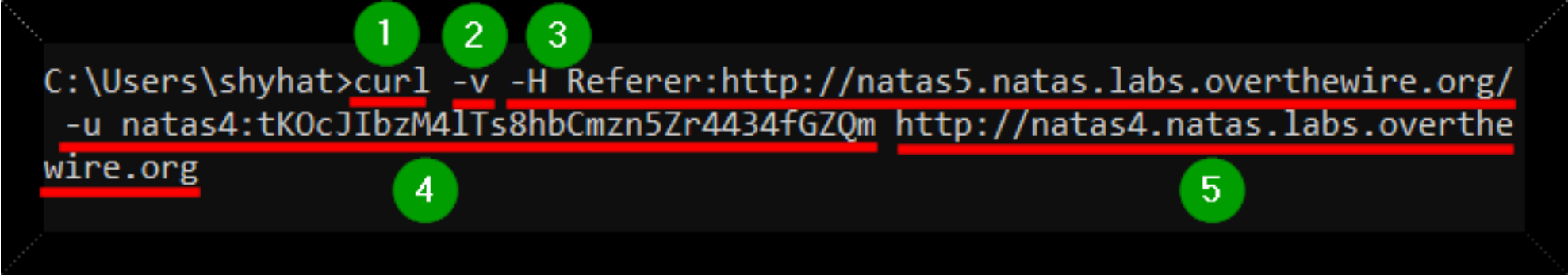
The cURL Program

```
C:\Users\shyhat>curl -v -H Referer:http://natas5.natas.labs.overthewire.org/ -u
natas4:tK0cJIbzM4lTs8hbCmzn5Zr4434fGZQm http://natas4.natas.labs.overthewire.org

* Trying 13.50.142.37:80...
* Connected to natas4.natas.labs.overthewire.org (13.50.142.37) port 80 (#0)
* Server auth using Basic with user 'natas4'
> GET / HTTP/1.1
> Host: natas4.natas.labs.overthewire.org
> Authorization: Basic bmF0YXM0OnRLT2NKSWJ6TTRsVHM4aGJDbXpuNVpyNDQzNGZHWlFt
```

This app allows for modification of various HTTP variables, which normal web browsers are not capable of, unless modified.

The cURL Program



```
C:\Users\shyhat>curl -v -H Referer:http://natas5.natas.labs.overthewire.org/  
-u natas4:tK0cJIbzM4lTs8hbCmzn5Zr4434fGZQm http://natas4.natas.labs.overthe  
wire.org
```

- 1 – The command itself
- 2 – The verbose output switch
- 3 – The HTTP header argument
- 4 – The user authentication argument
- 5 – The webpage to be accessed

P-5 HTTP Cookie Header

Another extremely common HTTP header is the Cookie header, which is used to retain user settings or establish / maintain a user session on a website.



P-5 HTTP Cookie Header

For example, a website has a button on its user preferences page which sets the webpage background color for the website.



P-5 HTTP Cookie Header

Once the color is selected, the web server will send a Cookie to the web browser to be used anytime the website is visited, changing the webpage's background colors to whatever is specified in the Cookie.



P-5 HTTP Cookie Header

Similarly, when a user successfully logs into a website, the web server will send the web browser a Cookie that identifies which user session is being used, and the browser will use that Cookie each time that website is accessed.



P-5 HTTP Cookie Header

Any Cookie that is used for user sessions has the potential for security abuse, so it is important that the Cookie values created for user sessions are not predictable at all.



HTTP Methods

```
C:\Users\User>curl -vv -X POST https://example.com
* Host example.com:443 was resolved.
* IPv6: (none)
* IPv4: 93.184.215.14
*   Trying 93.184.215.14:443...
* Connected to example.com (93.184.215.14) port 443
* schannel: disabled automatic use of client certificate
* ALPN: curl offers http/1.1
* ALPN: server accepted http/1.1
* using HTTP/1.x
> POST / HTTP/1.1
> Host: example.com
> User-Agent: curl/8.8.0
```

All HTTP requests are made with an HTTP method, sometimes called an HTTP verb.

HTTP Methods

```
C:\Users\User>curl -vv -X POST https://example.com
* Host example.com:443 was resolved.
* IPv6: (none)
* IPv4: 93.184.215.14
*   Trying 93.184.215.14:443...
* Connected to example.com (93.184.215.14) port 443
* schannel: disabled automatic use of client certificate
* ALPN: curl offers http/1.1
* ALPN: server accepted http/1.1
* using HTTP/1.x
> POST / HTTP/1.1
> Host: example.com
> User-Agent: curl/8.8.0
```

Webpages send the browser different content depending on which HTTP method is used.

HTTP Methods

```
C:\Users\User>curl -vv -X POST https://example.com
* Host example.com:443 was resolved.
* IPv6: (none)
* IPv4: 93.184.215.14
*   Trying 93.184.215.14:443...
* Connected to example.com (93.184.215.14) port 443
* schannel: disabled automatic use of client certificate
* ALPN: curl offers http/1.1
* ALPN: server accepted http/1.1
* using HTTP/1.x
> POST / HTTP/1.1
> Host: example.com
> User-Agent: curl/8.8.0
```

The most common HTTP methods are the GET and POST methods.

HTTP Request Methods



GET

Retrieves data or resource from a specified URL. It is used to retrieve information without modifying it.



POST

Submit data or creates a new resource on the server. It is used to send data to be processed by the server. It often results in the creation of a new resource on the server.



PUT

Updates the existing resource with the new data. It replaces the entire resource or creates it if it does not exist.



DELETE

Deletes the specified resource from the server.



PATCH

Partially updates the existing resource with the provided data. PATCH request does not create a new resource if the specified resource does not exist on the server.



HEAD

Retrieve only the headers of a response. It is used to check the status or headers of a resource without fetching the actual content.



OPTIONS

Fetch or Retrieve the allowed methods and other information of the specified resource.



TRACE

The TRACE method echoes back the received request to the client, allowing clients to inspect the request and see any modifications or additions made by intermediaries. It is mainly used for the diagnostic purposes.



CONNECT

Converts the request connection to a transparent TCP/IP tunnel, commonly used for establishing secure SSL/TLS connections through proxies.

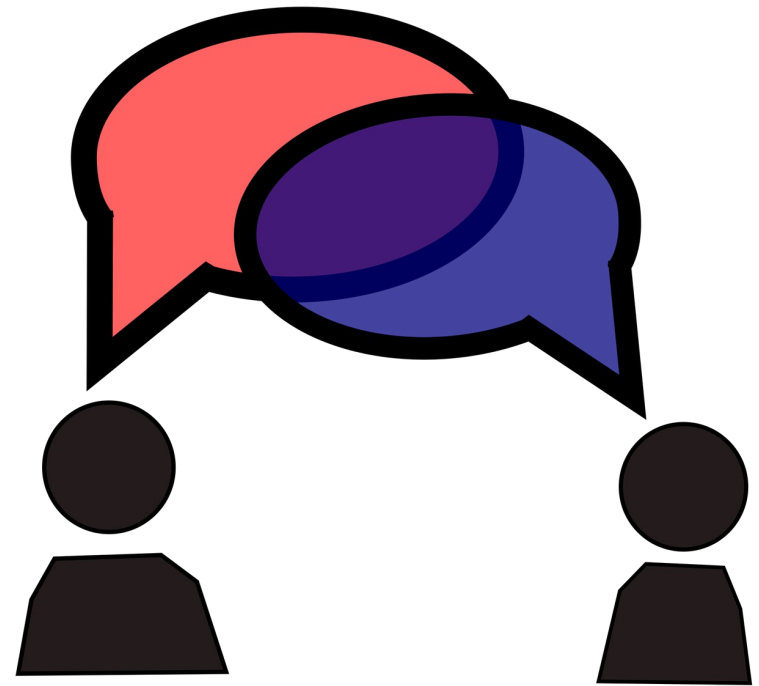
Summary



Let's review the web exploitation concepts we learned in today's workshop:

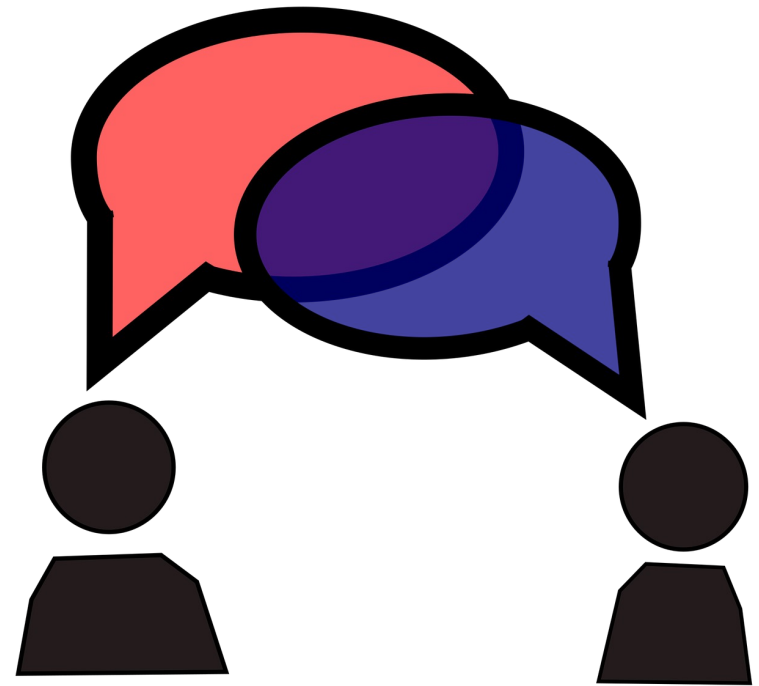
HTTP Referer Header

The HTTP Referer header (which is misspelled on purpose) contains the value of a complete or partial address of the webpage that is making the request. It's often used for analytics or logging.



HTTP Referer Header

The HTTP Referer header is occasionally used by web app developers as a security mechanism, but this is not a good practice, as it was never intended to be used in such a way



HTTP Cookie Header

HTTP Cookies are commonly used as a security mechanism that allows users to maintain an authenticated user session on a website.



HTTP Cookie Header

In short, HTTP cookies allow users to “login” to websites



HTTP Cookie Header

But since cookie values can be modified by the user, proper care should be taken to ensure that valid cookie values are not predictable or easily guessed



What's Next?

In the next HackerFrogs Afterschool web app hacking workshop, we'll be learning more web app hacking skills with Natas CTF.



Extra Credit

Looking for more study material on this workshop's topics?

See this video's description for links to supplemental documents and exercises!



Until Next Time, HackerFrogs!

