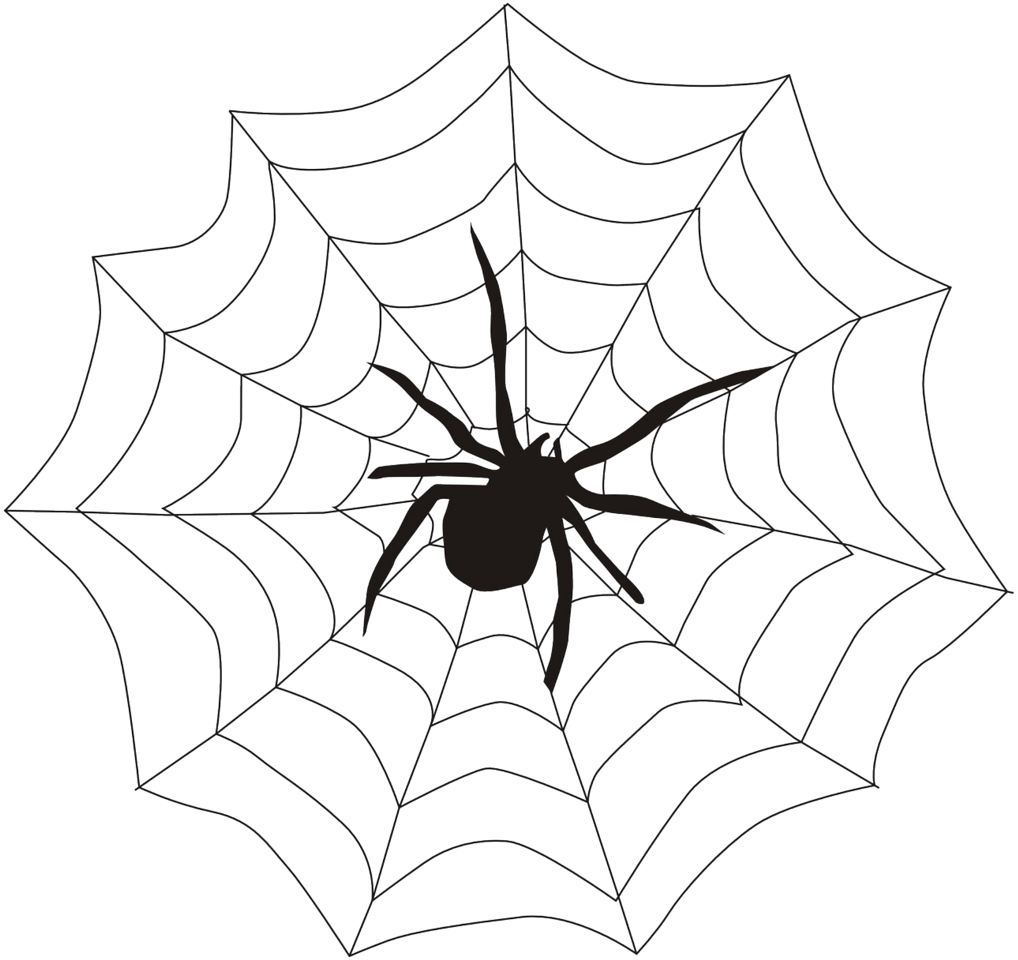
HackerFrogs Afterschool OverTheWire Natas: Part 2

Class:
Web App Hacking

Workshop Number:
AS-WEB-02

Document Version:
1.2

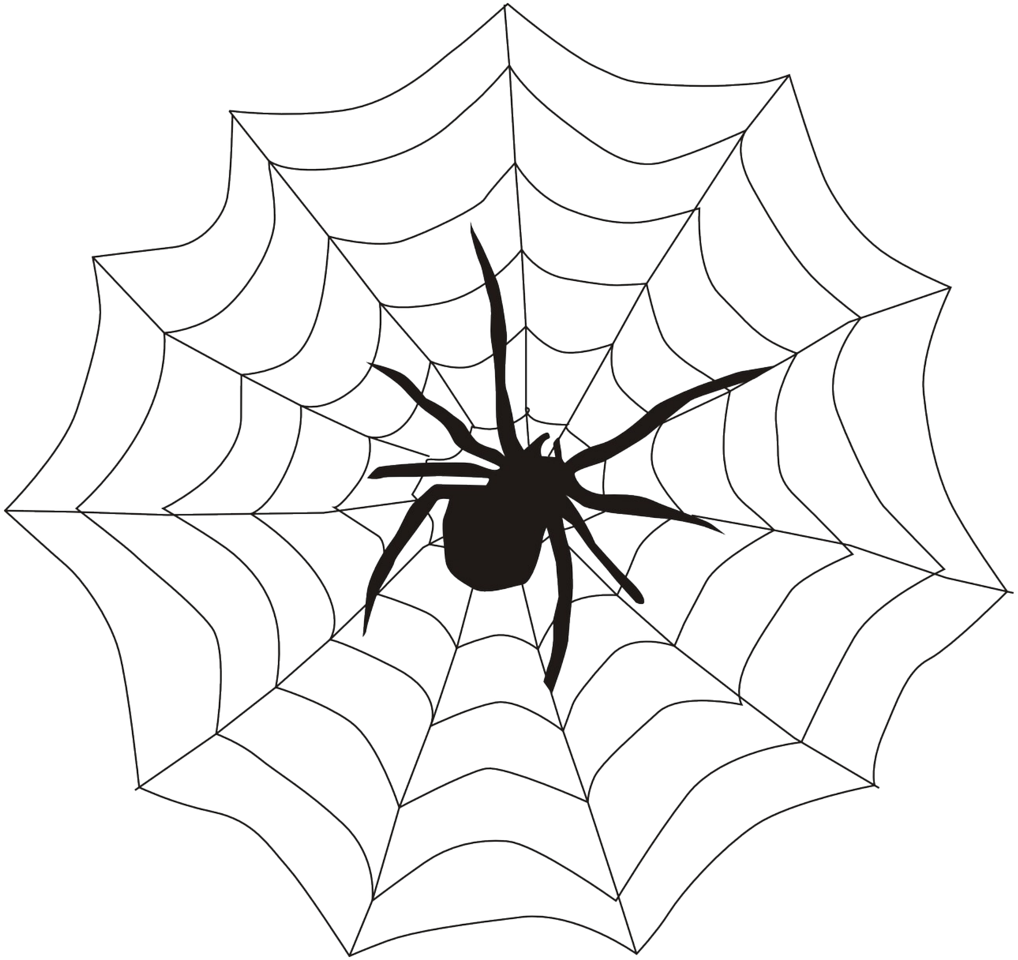
Special Requirements:
None



Web App Hacking

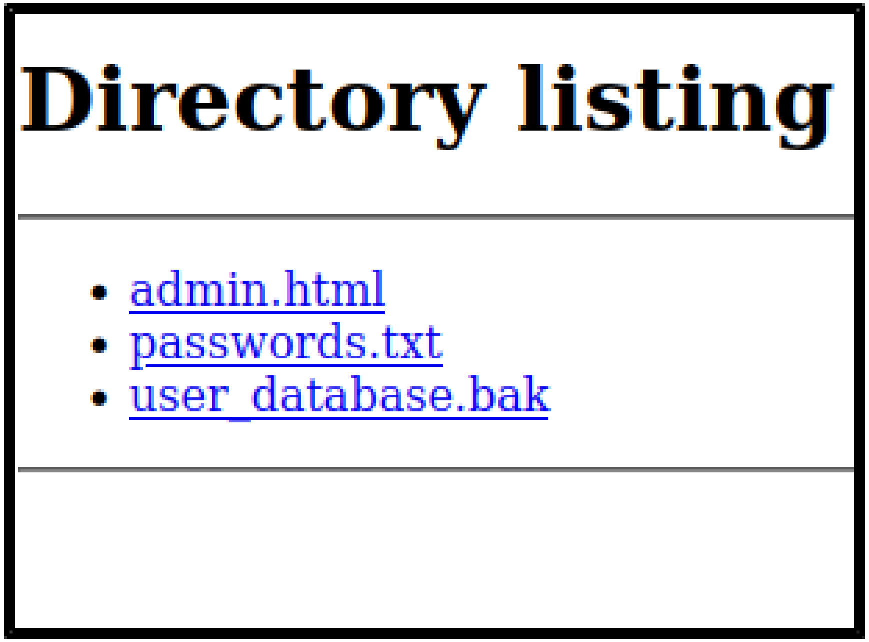
This is the second workshop in our intro to web app hacking course.

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.

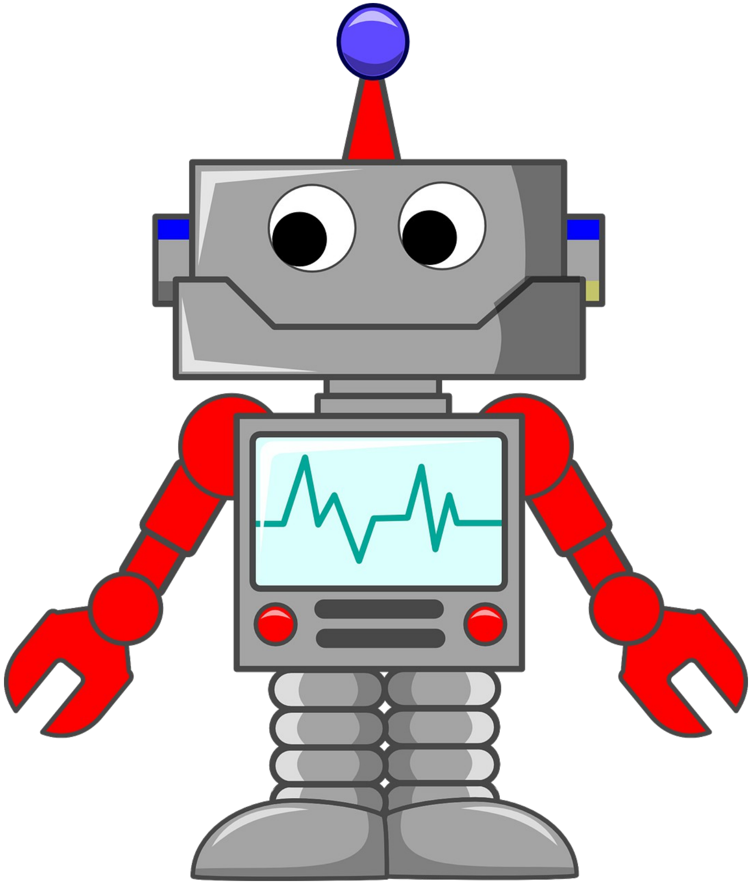


Directory listing

- 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., 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., 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., 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., natas4.overthewire.org

`User-Agent` ← the type of browser that is making the request
e.g., Chrome/0.2

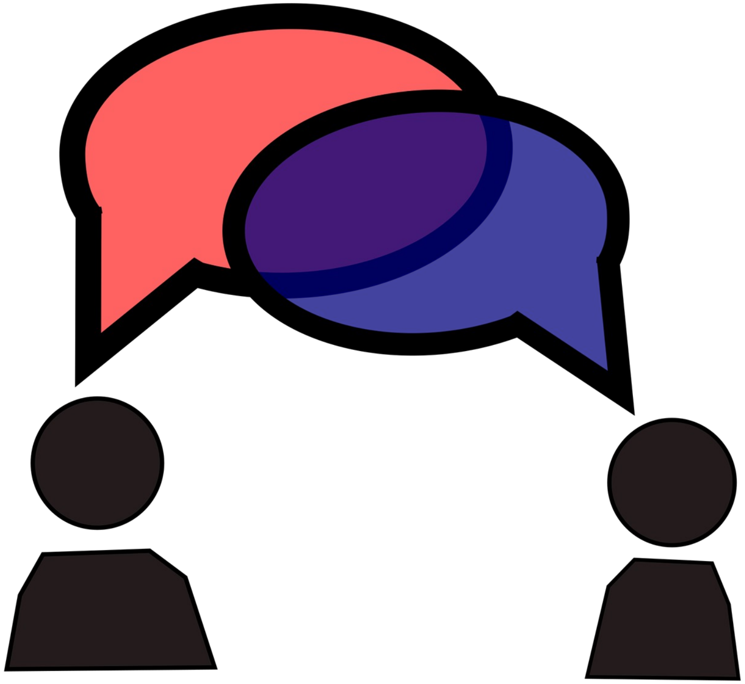
`Accept` ← the type of data that should be sent in response
e.g., */* (any type of data)

P-4 HTTP Headers

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.

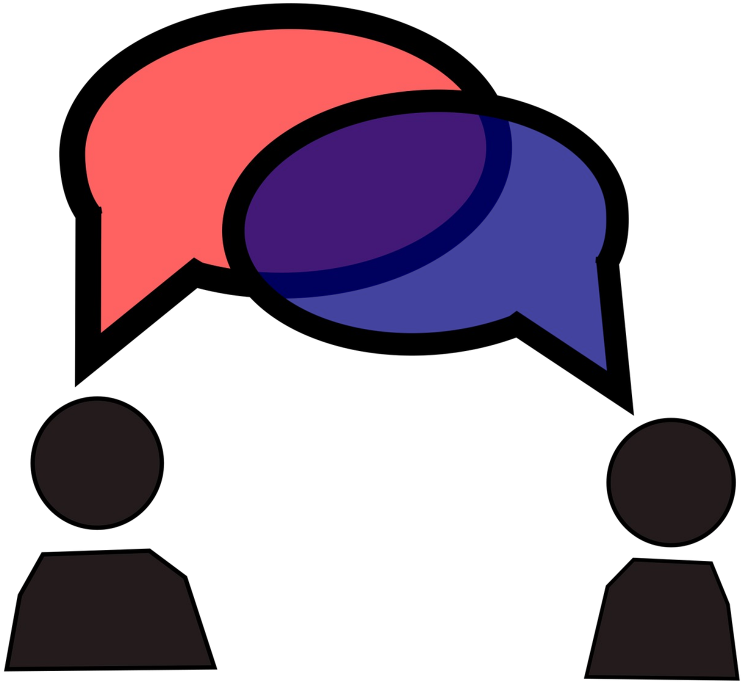
P-4 HTTP Referer Header

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.



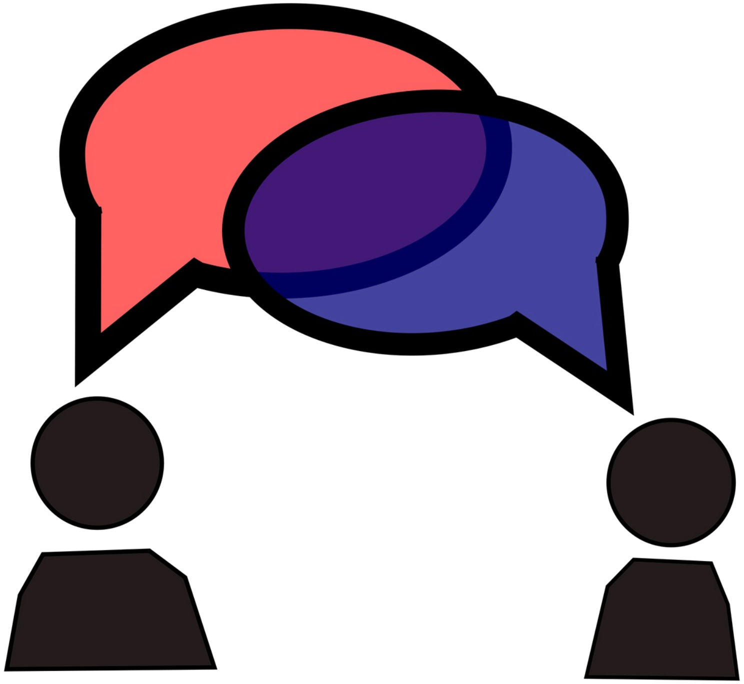
P-4 HTTP Referer Header

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



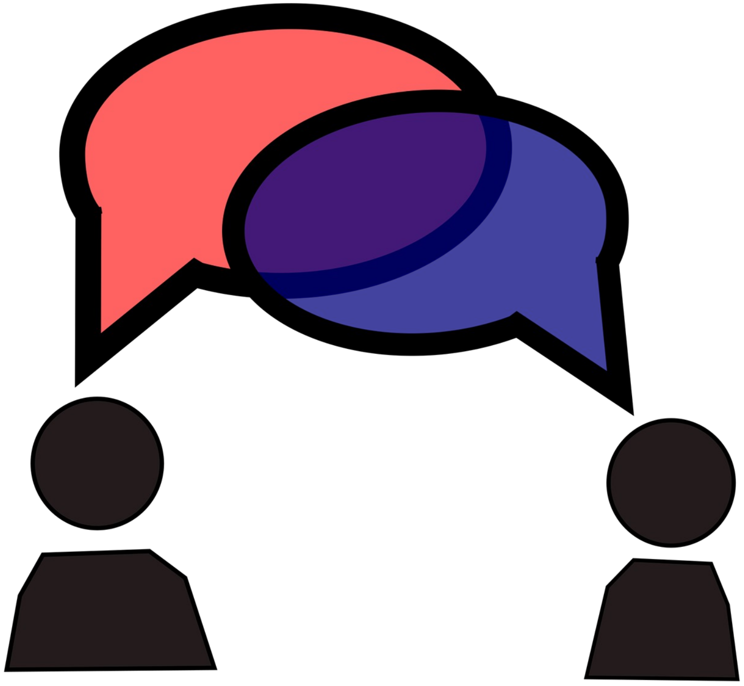
P-4 HTTP Referer Header

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

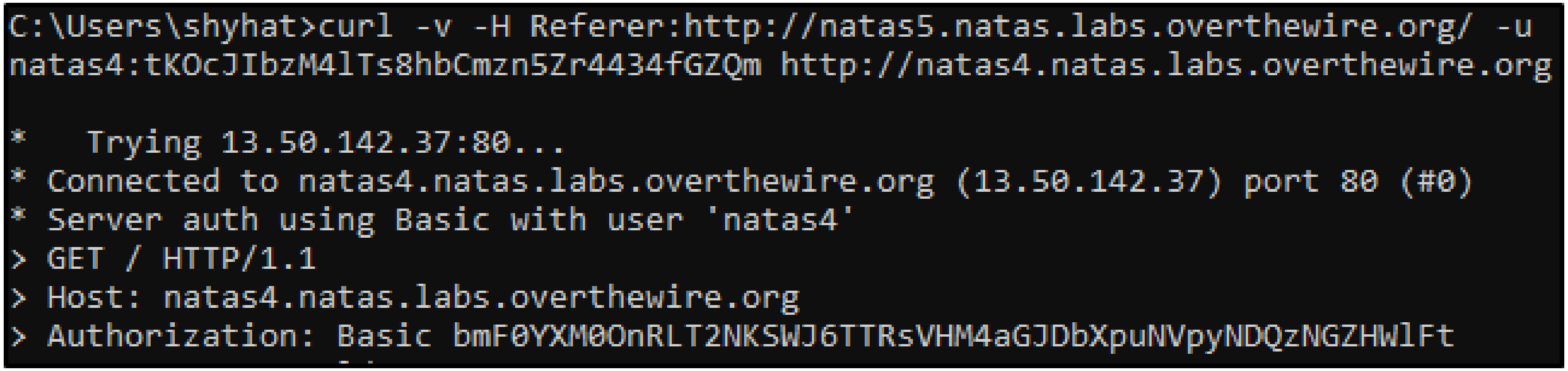


P-4 HTTP Referer Header

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



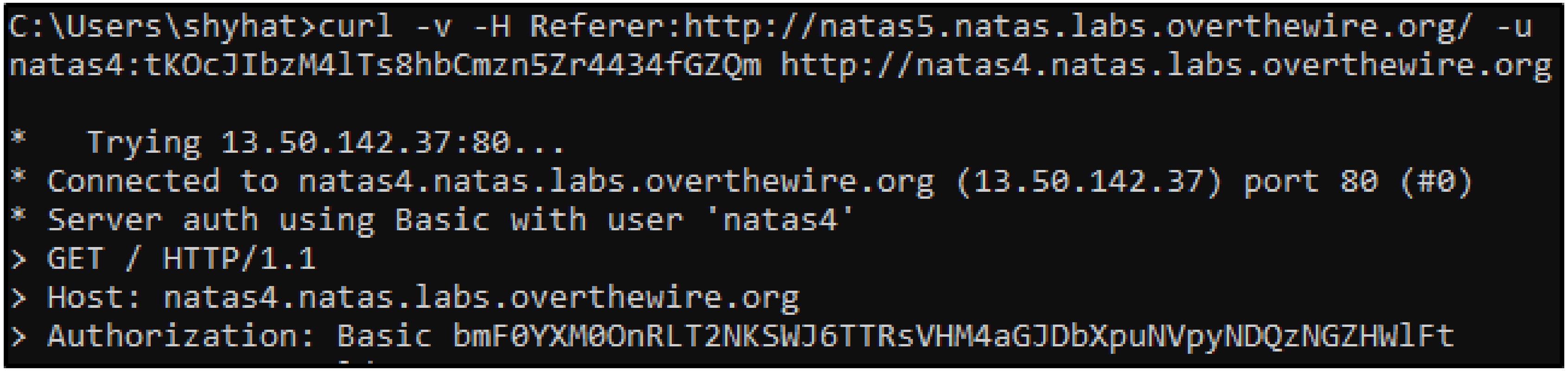
The cURL Program



```
C:\Users\shyhat>curl -v -H Referer:http://natas5.natas.labs.overthewire.org/ -u
natas4:tKOcJIbzM4lTs8hbCmzn5Zr4434fGZQm 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 cURL program is a command line interface (CLI) app that is common to all major computer operating systems.

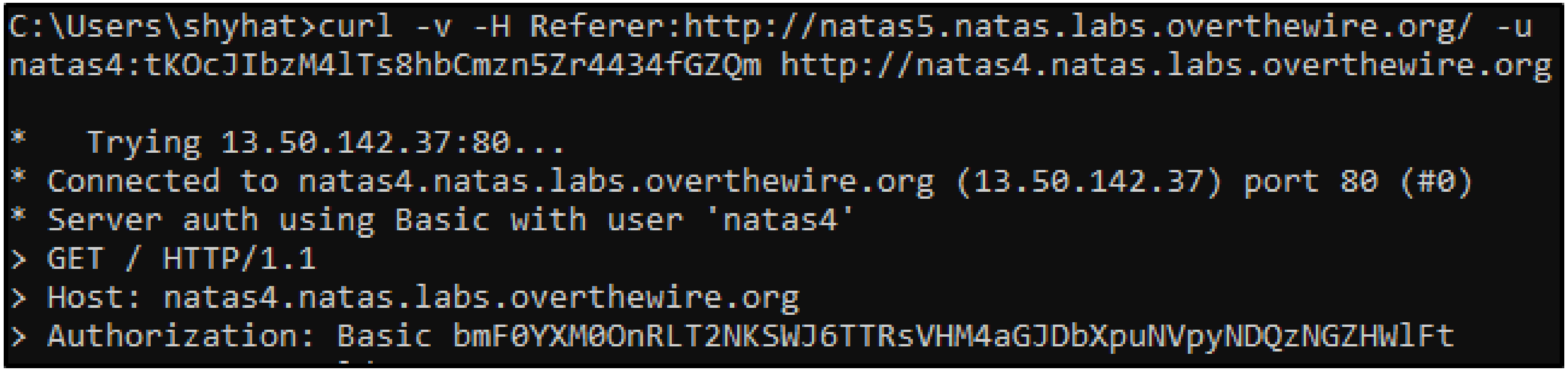
The cURL Program



```
C:\Users\shyhat>curl -v -H Referer:http://natas5.natas.labs.overthewire.org/ -u
natas4:tKOcJIbzM4lTs8hbCmzn5Zr4434fGZQm 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.

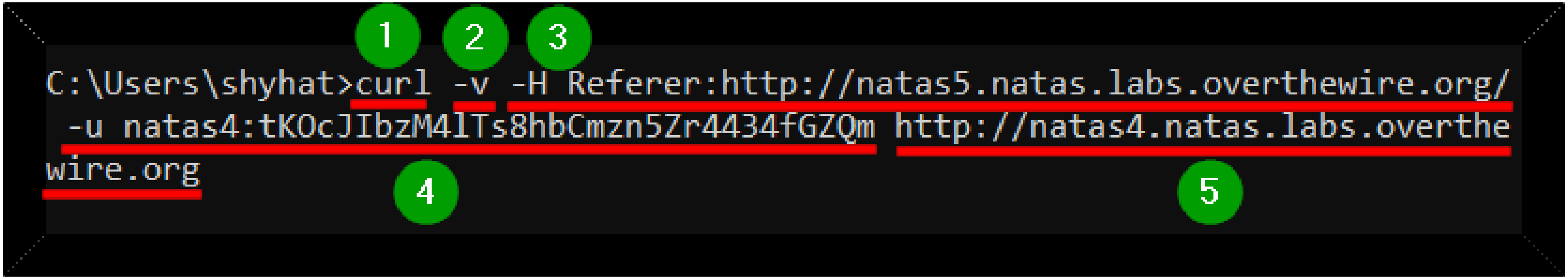
The cURL Program



```
C:\Users\shyhat>curl -v -H Referer:http://natas5.natas.labs.overthewire.org/ -u
natas4:tKOcJIbzM4lTs8hbCmzn5Zr4434fGZQm 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:tKOcJIbzM4lTs8hbCmzn5Zr4434fGZQm http://natas4.natas.labs.overthe
wire.org
```

1 – The command itself

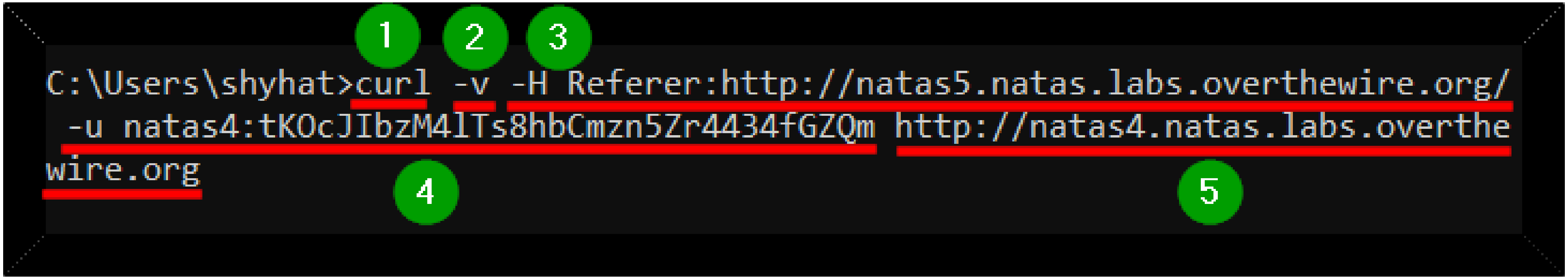
2 – The verbose output switch

3 – The HTTP header argument

4 – The user authentication argument

5 – The webpage to be accessed

The cURL Program



```
C:\Users\shyhat>curl -v -H Referer:http://natas5.natas.labs.overthewire.org/
 -u natas4:tKOcJIbzM4lTs8hbCmzn5Zr4434fGZQm http://natas4.natas.labs.overthe
wire.org
```

1 – The command itself

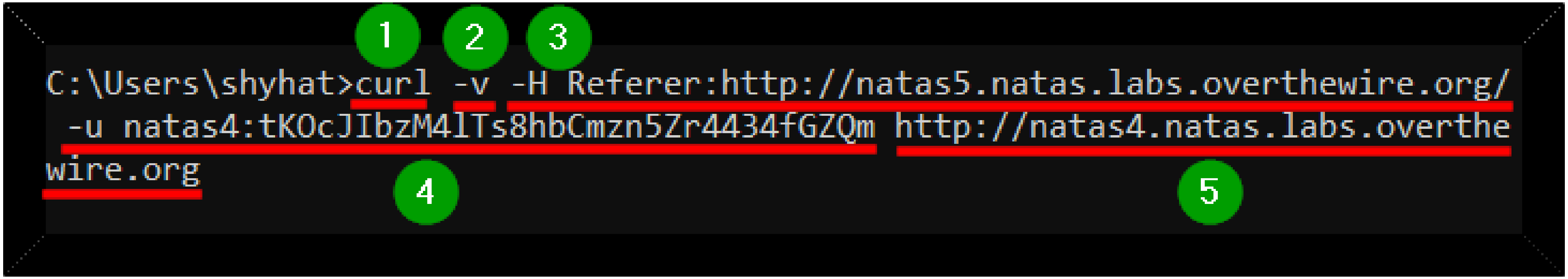
2 – The verbose output switch

3 – The HTTP header argument

4 – The user authentication argument

5 – The webpage to be accessed

The cURL Program



```
C:\Users\shyhat>curl -v -H Referer:http://natas5.natas.labs.overthewire.org/
 -u natas4:tKOcJIbzM4lTs8hbCmzn5Zr4434fGZQm http://natas4.natas.labs.overthe
wire.org
```

1 – The command itself

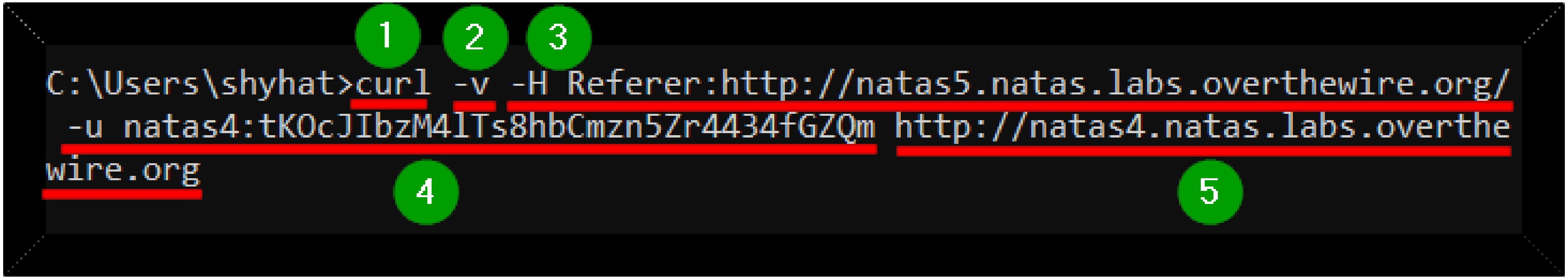
2 – The verbose output switch

3 – The HTTP header argument

4 – The user authentication argument

5 – The webpage to be accessed

The cURL Program



```
C:\Users\shyhat>curl -v -H Referer:http://natas5.natas.labs.overthewire.org/
 -u natas4:tKOcJIbzM4lTs8hbCmzn5Zr4434fGZQm http://natas4.natas.labs.overthe
wire.org
```

1 – The command itself

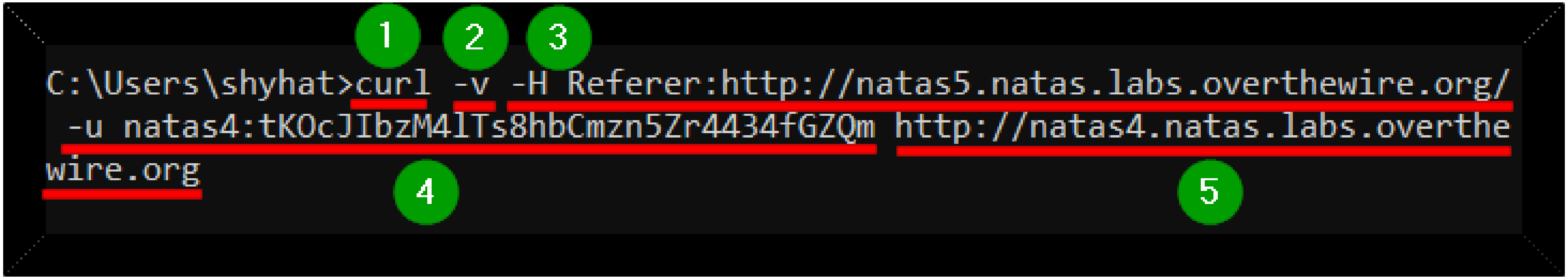
2 – The verbose output switch

3 – The HTTP header argument

4 – The user authentication argument

5 – The webpage to be accessed

The cURL Program



```
C:\Users\shyhat>curl -v -H Referer:http://natas5.natas.labs.overthewire.org/
 -u natas4:tKOcJIbzM4lTs8hbCmzn5Zr4434fGZQm http://natas4.natas.labs.overthe
wire.org
```

1 – The command itself

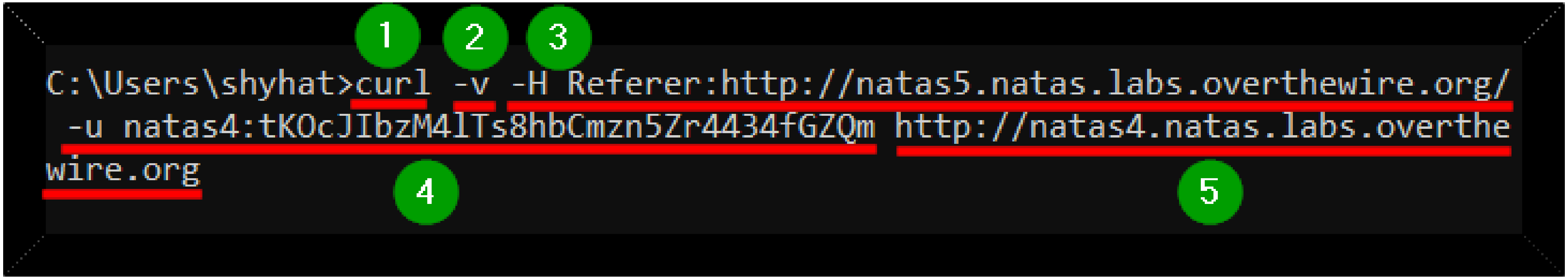
2 – The verbose output switch

3 – The HTTP header argument

4 – The user authentication argument

5 – The webpage to be accessed

The cURL Program



```
C:\Users\shyhat>curl -v -H Referer:http://natas5.natas.labs.overthewire.org/
 -u natas4:tKOcJIbzM4lTs8hbCmzn5Zr4434fGZQm 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

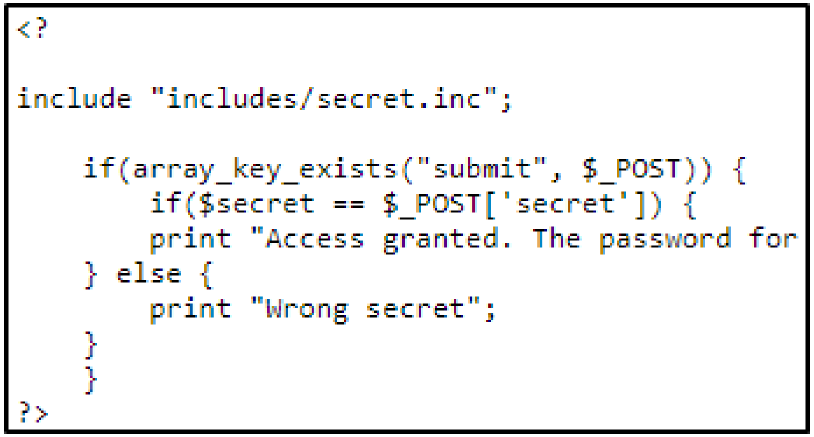
Similarly, when a user successfully logs into a website, the web server will send the web browser a Cookie that identifies which user session is being used, and the browser will use that Cookie each time that website is accessed.

P-5 HTTP Cookie Header

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

P-6 Sourcecode Analysis

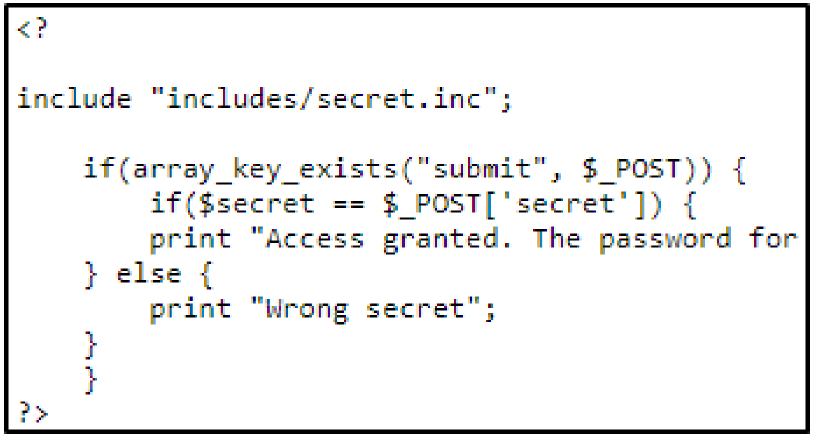
Sourcecode analysis is the process of analyzing the code of a piece of software with the goal of deeper understanding regarding its function.



```
<?

include "includes/secret.inc";

    if(array_key_exists("submit", $_POST)) {
        if($secret == $_POST['secret']) {
        print "Access granted. The password for
    } else {
        print "Wrong secret";
    }
    }
?>
```

P-6 Sourcecode Analysis

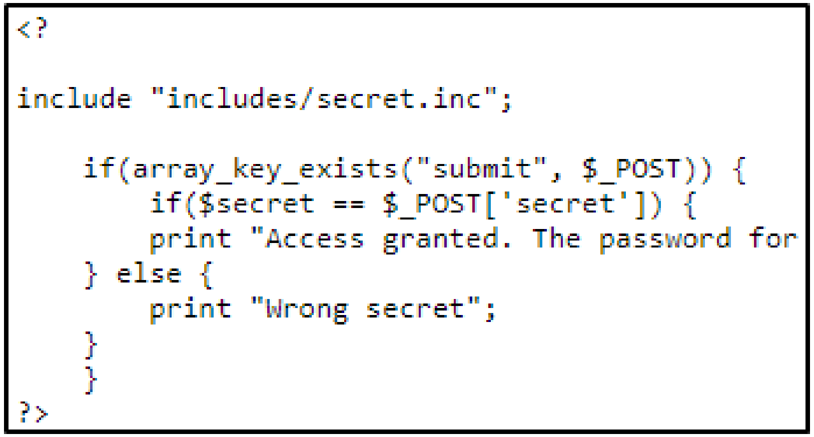
In the context of web exploitation, the code involved is usually written in HTML, CSS, JavaScript, PHP, or Java.



```
<?

include "includes/secret.inc";

    if(array_key_exists("submit", $_POST)) {
        if($secret == $_POST['secret']) {
        print "Access granted. The password for
    } else {
        print "Wrong secret";
    }
    }
?>
```

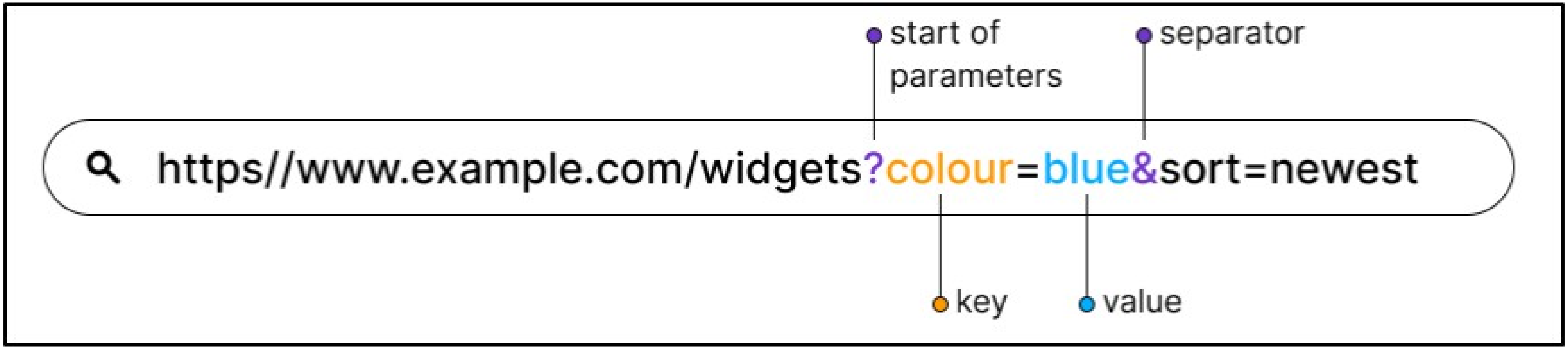
P-6 Sourcecode Analysis

Therefore, successful sourcecode analysis requires a certain level of familiarity in the language the code is written in.



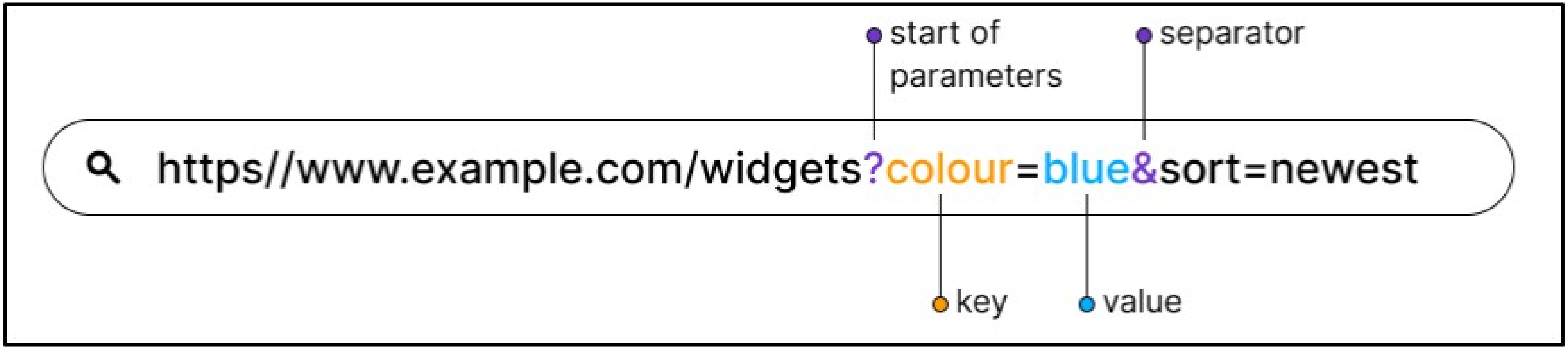
```
<?

include "includes/secret.inc";

    if(array_key_exists("submit", $_POST)) {
        if($secret == $_POST['secret']) {
        print "Access granted. The password for
    } else {
        print "Wrong secret";
    }
    }
?>
```

P-7 URL Parameters



URL parameters are variables attached to the end of URLs. They can be identified by the ? (question mark) directly after the webpage or directory name, followed by the parameter key name, then the = (equals) sign, then the value.

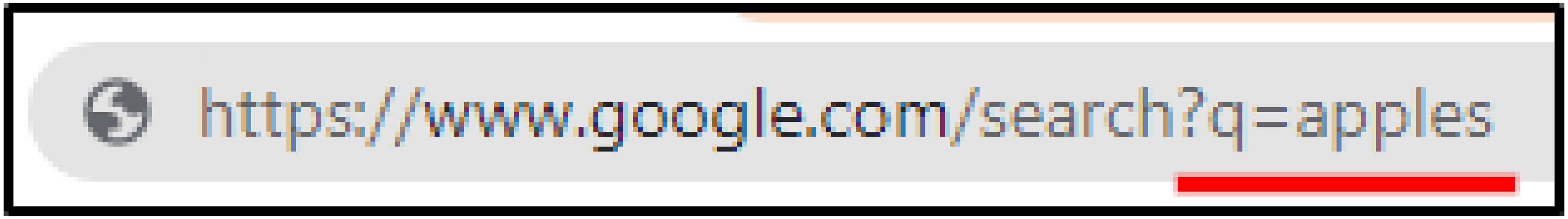
P-7 URL Parameters



https//www.example.com/widgets?colour=blue&sort=newest

If there are multiple parameters included in the same URL, then they are separated by the & (ampersand) symbol.

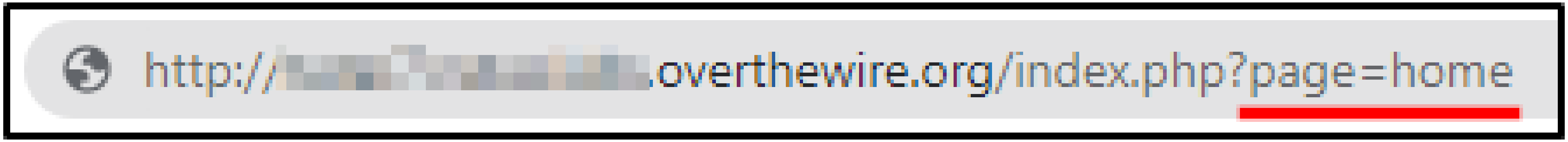
P-7 URL Parameters Use Cases



https://www.google.com/search?q=apples

There are a few different reasons why webpages use URL parameters. The most common one is for search queries.

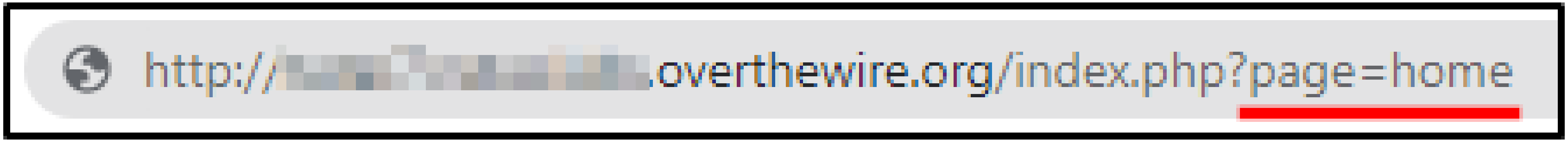
P-7 URL Parameters Use Cases



http://[redacted].overthewire.org/index.php?page=home

However, another common, and potentially dangerous use of URL parameters is to instruct the webserver on which webpage to display.

P-7 URL Parameters Use Cases



http://[redacted].overthewire.org/index.php?page=home

The use of URL parameters which reference other files on the webserver could potentially be exploited in an attack called Local File Inclusion (LFI).

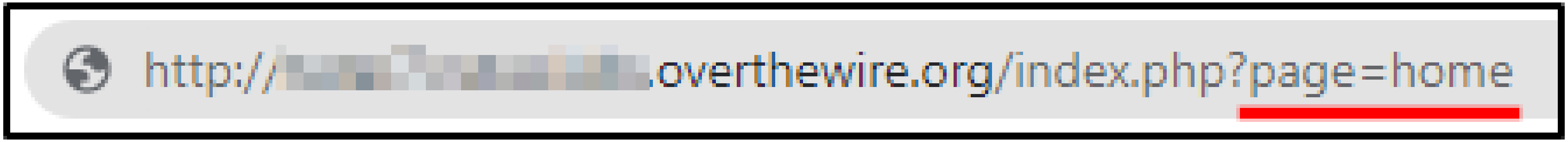
P-7 Local File Inclusion

Local File Inclusion (LFI), aka Directory Traversal, or Path Traversal, is a web app vulnerability where arbitrary local webserver files can be accessed through a web interface.

P-7 Local File Inclusion

LFI vulnerabilities can lead to sensitive data exposure, and can also be used as the first step in a chain of exploits.

P-7 Local File Inclusion



http://[redacted].overthewire.org/index.php?page=home

The inclusion of file names in URL parameters is a typical method through which a potential LFI vulnerability is identified.

P-7 Local File Inclusion: Filesystem Structure

Each **../** indicates an elevation of one level in the filesystem, traveling from the web app's working directory (/natas7) up to the top-level directory (/)

/
/var
/var/html
/var/html/labs
/var/html/labs/natas
/var/html/labs/natas/natas7

P-7 Local File Inclusion: Filesystem Structure

From the top-level directory, we can provide a filepath to the file we want to access.

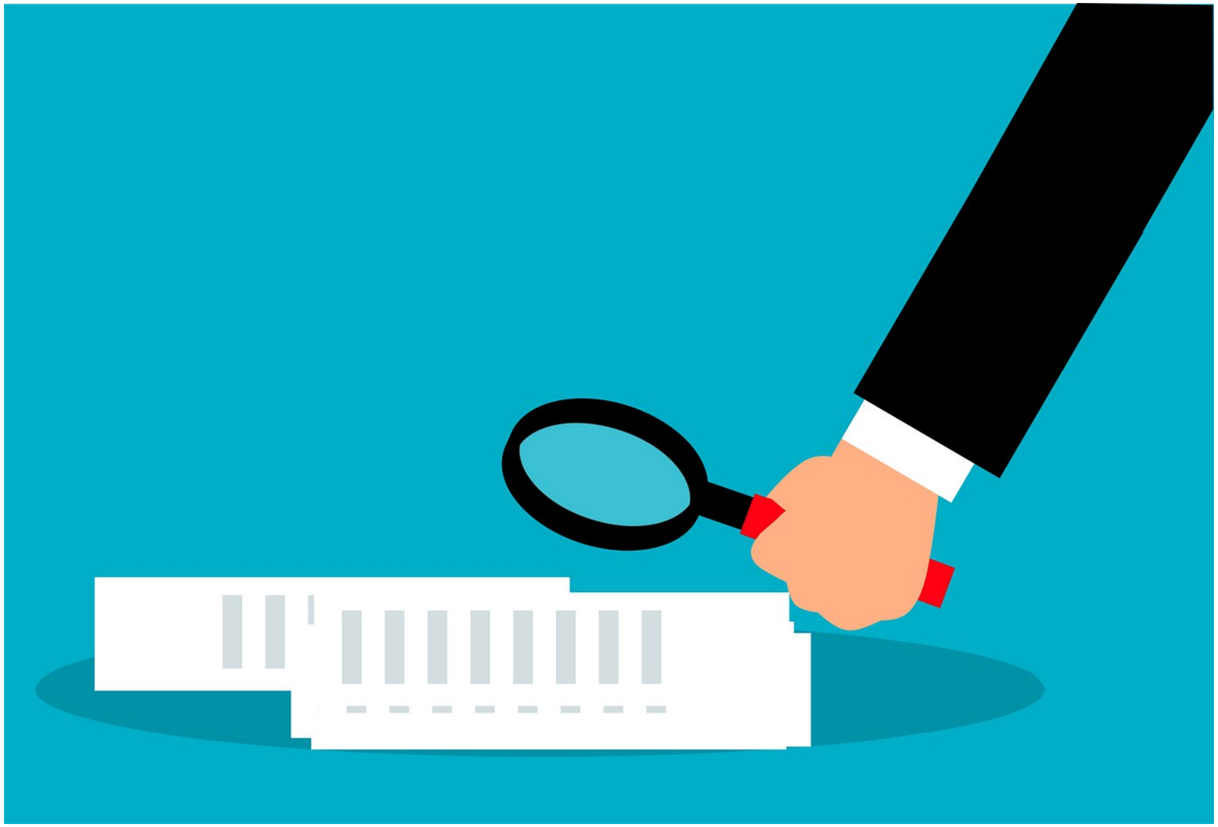
A typical test file for LFI on Linux / Unix webservers is the **/etc/passwd** file, since it is publicly readable by default, and gives info regarding usernames on the webserver.

P-7 Local File Inclusion: Web App File Access

Web apps cannot access files on the webserver that are on a higher-level directory than the app's working directory, which is usually located several directories down from the server's top-level directory.

In an LFI attack, a series of ../ are used to escape the app's working directory to access other files on the webserver.

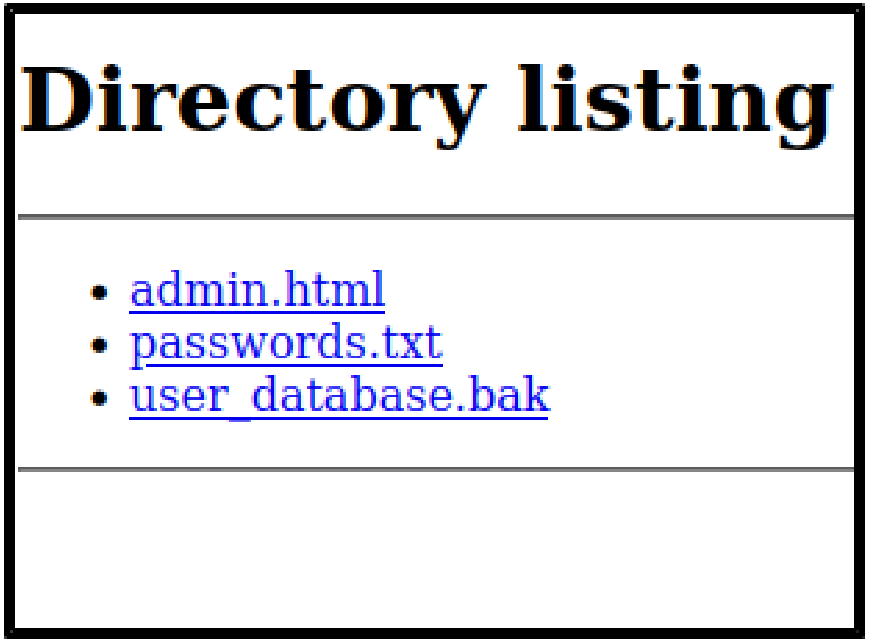
Summary



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

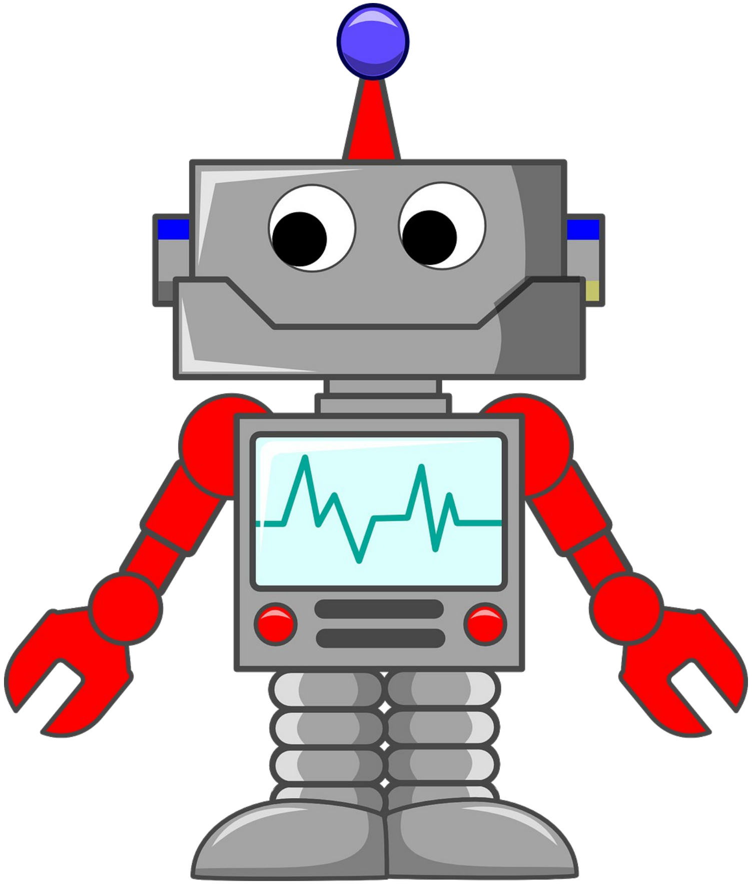


Directory listing

- admin.html
- passwords.txt
- user_database.bak

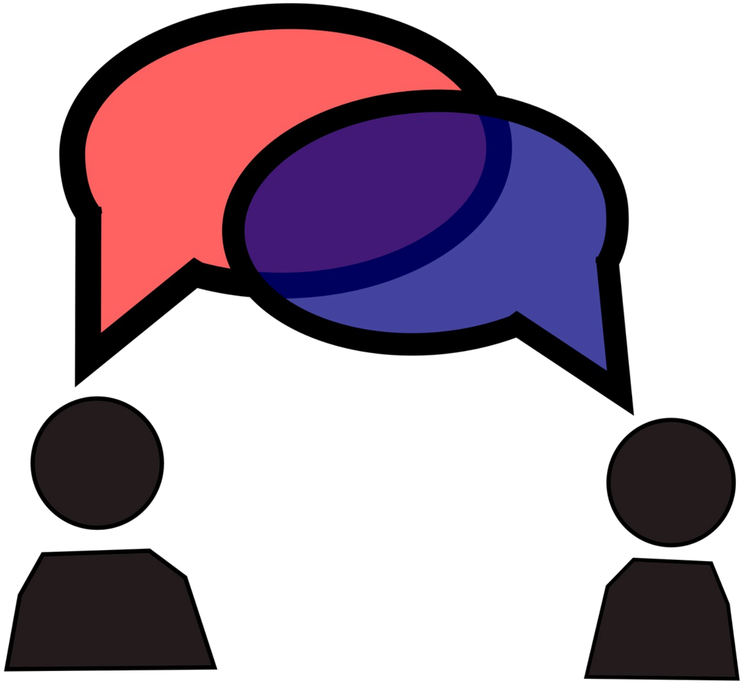
Robots.txt

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.



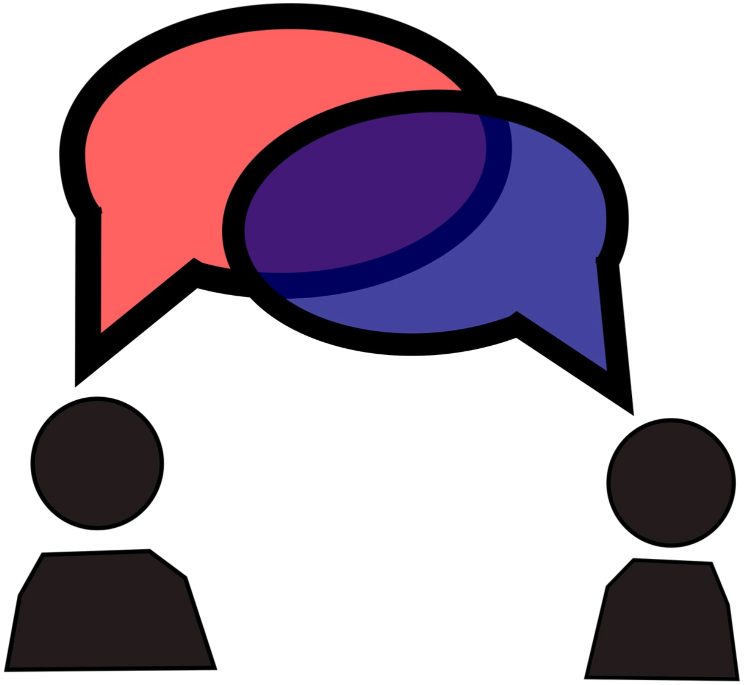
HTTP Referer Header

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 conclude our time learning 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!