

COMP 307-LabAssignment#5

Student Name : Najmun Nahar

Student Number : 301160081

In this lab we will do the following:

A. Detect a hacking attack using a PHPIDS

1) Enable PHPIDS **Take a snapshot**

The screenshot shows the 'security.php' page of the Damn Vulnerable Web Application (DVWA). The left sidebar contains a menu with options: File Inclusion, File Upload, Insecure CAPTCHA, SQL Injection, SQL Injection (Blind), Weak Session IDs, XSS (DOM), XSS (Reflected), XSS (Stored), CSP Bypass, JavaScript, DVWA Security (highlighted), PHP Info, About, and Logout. The main content area is titled 'PHPIDS' and describes it as a security layer for PHP-based web applications. It lists four difficulty levels: 1. Low, 2. Medium, 3. High, and 4. Impossible. The 'Impossible' level is selected in a dropdown menu, and the 'Submit' button is visible. Below the dropdown, a message states 'PHPIDS is now enabled'. At the bottom of the page, a status bar displays 'Damn Vulnerable Web Application (DVWA) v1.10 "Development"'.

localhost/DVWA-master/security.php

File Inclusion
File Upload
Insecure CAPTCHA
SQL Injection
SQL Injection (Blind)
Weak Session IDs
XSS (DOM)
XSS (Reflected)
XSS (Stored)
CSP Bypass
JavaScript
DVWA Security
PHP Info
About
Logout

as a platform to teach or learn basic exploitation techniques.
2. Medium - This setting is mainly to give an example to the user of **bad security practices**, where the developer has tried but failed to secure an application. It also acts as a challenge to users to refine their exploitation techniques.
3. High - This option is an extension to the medium difficulty, with a mixture of **harder or alternative bad practices** to attempt to secure the code. The vulnerability may not allow the same extent of the exploitation, similar in various Capture The Flags (CTFs) competitions.
4. Impossible - This level should be **secure against all vulnerabilities**. It is used to compare the vulnerable source code to the secure source code.
Prior to DVWA v1.9, this level was known as 'high'.

Impossible Submit

PHPIDS

PHPIDS v0.6 (PHP-Intrusion Detection System) is a security layer for PHP based web applications.

PHPIDS works by filtering any user supplied input against a blacklist of potentially malicious code. It is used in DVWA to serve as a live example of how Web Application Firewalls (WAFs) can help improve security and in some cases how WAFs can be circumvented.

You can enable PHPIDS across this site for the duration of your session.

PHPIDS is currently: **enabled**. [\[Disable PHPIDS\]](#)

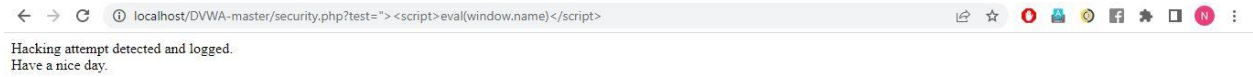
[\[Simulate attack\]](#) - [\[View IDS log\]](#)

PHPIDS is now enabled

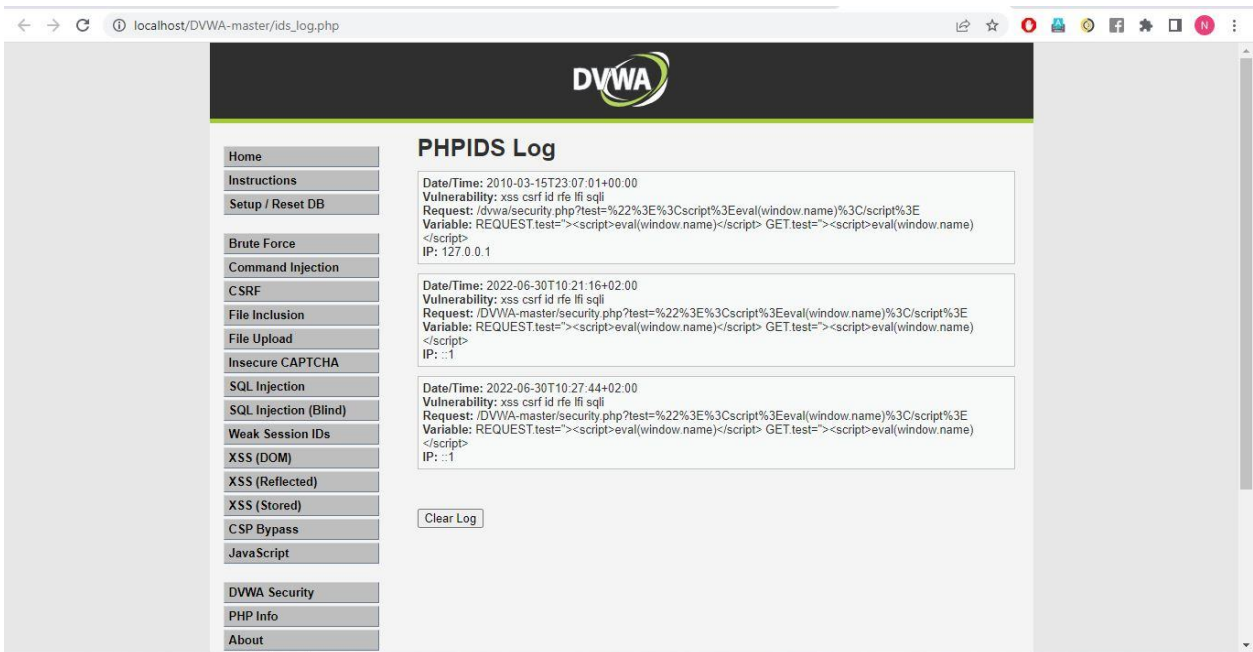
Username: admin
Security Level: impossible
Locale: en
PHPIDS: enabled
SQLi DB: mysql

Damn Vulnerable Web Application (DVWA) v1.10 "Development"

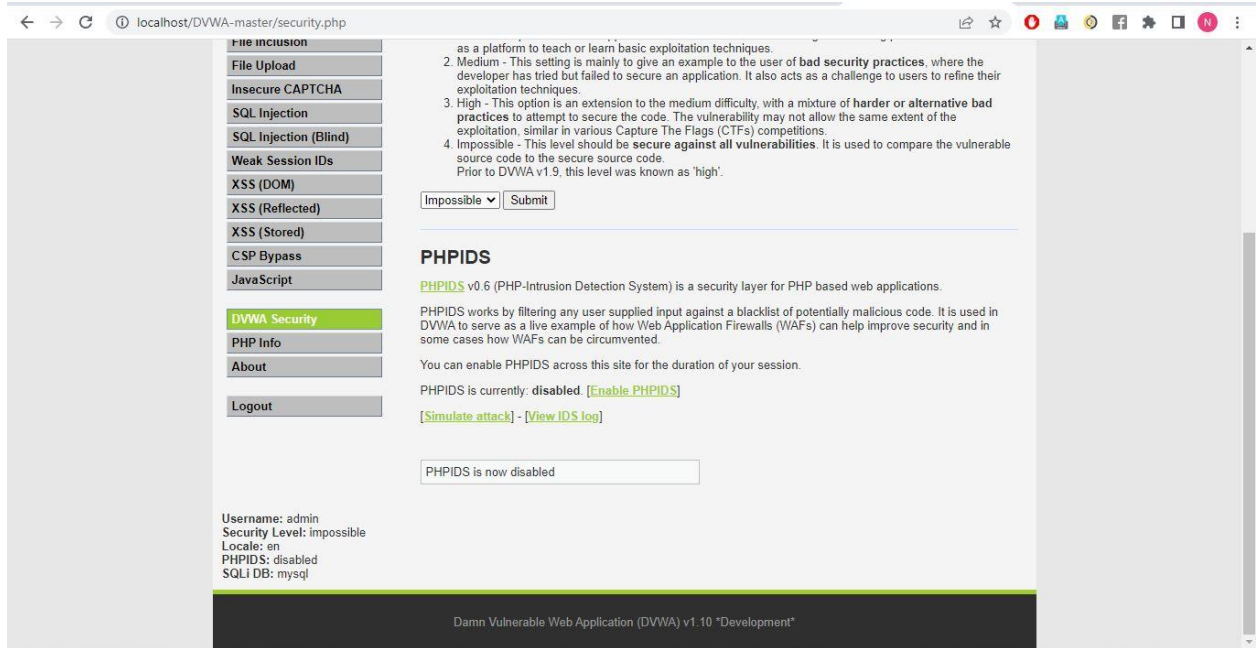
2) Simulate Attack



3) See log & take snapshot



4) Disable PHPIDS



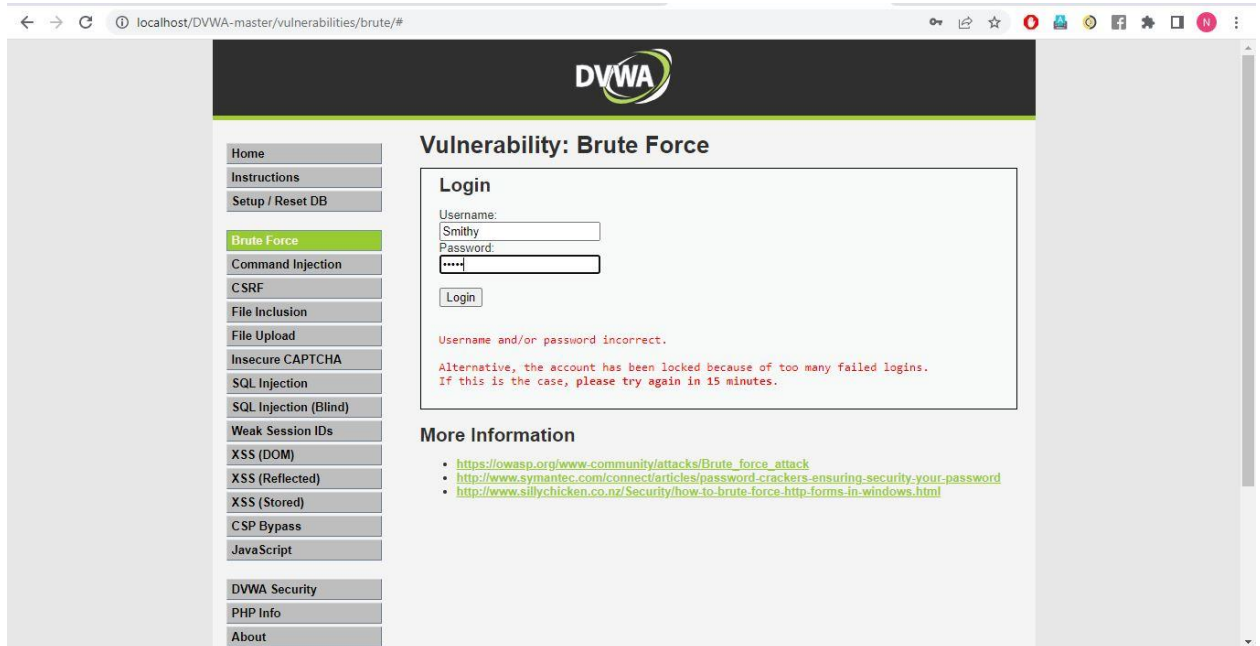
B. Brute Force

In the next steps, you will perform a brute force attack on the DVWA. In a brute force attack, a black-hat hacker literally breaks into the system by guessing the correct password. The brute force attack is a trial and error method for detecting username/password combinations often by using software that performs “dictionary attacks.” Dictionary attacks use exploits weak Dictionary common words found in a dictionary and passwords. Since dictionary attacks are most common, enforcing strong passwords and account lockouts can greatly diminish the risks. There are other ways to minimize risks and some of the most common include:

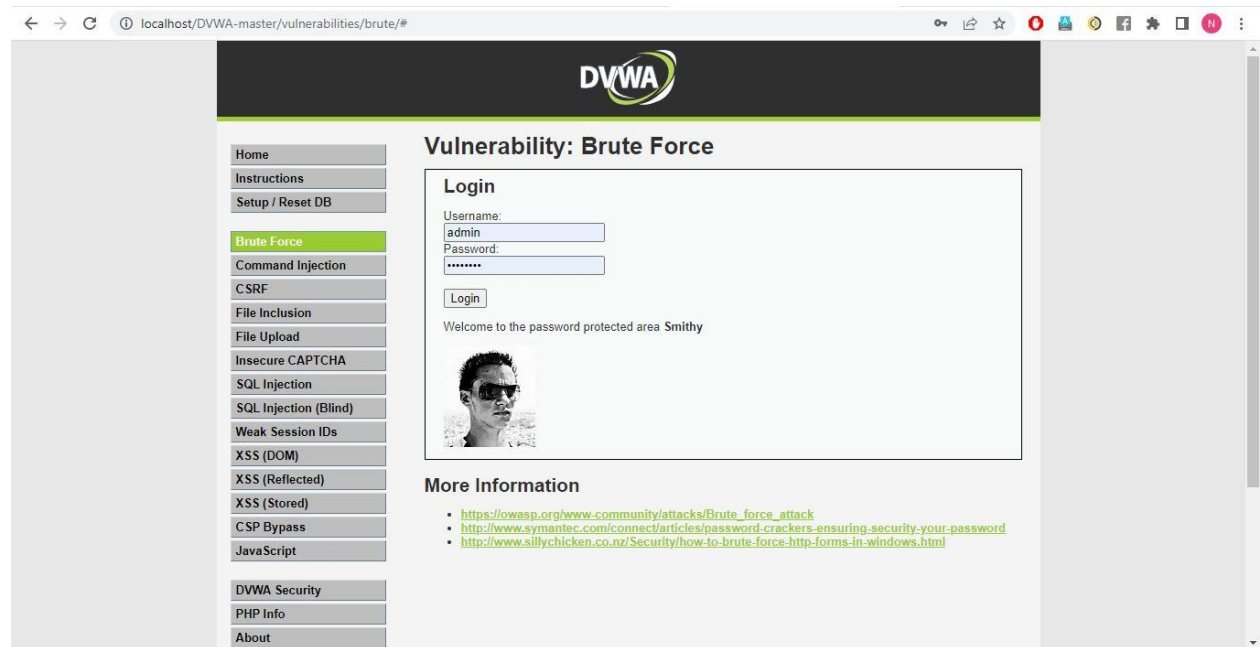
- Enforce strong password usage
- Use software that “locks” an account after a specific number of failed log on attempts
- Enable a Web application firewall that can detect brute force attacks and ban the offending IP address

1. In the DVWA navigation menu on the left, click the Brute Force button.

- On the Brute page, attempt a brute force login in DVWA using the following credentials and then press Login. Username: Smithy
Password: tryit
The DVWA tool will return an invalid username/password error. **Take a snapshot**



- Guess on the easy password for User Smithy and **take a snapshot**



4. Give the details for the differences between low, medium, high and impossible security for Brute Force attack step by step.

Low Level: In this credentials are passed as get parameters.

Medium Level: The medium level is basically identical except the implementation of sleep() functions to nag attackers.

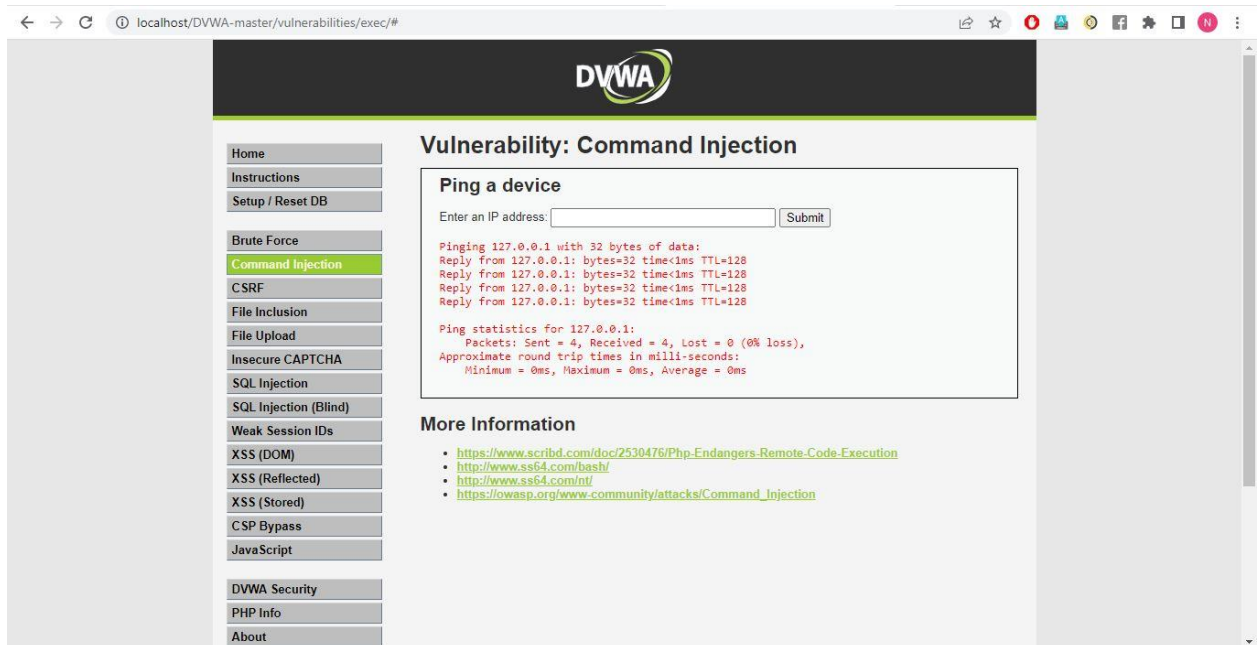
High Level: The high level is different. In this when we send a request with the credentials test/test a new parameter is sent along with the credentials: the user_token.

Impossible Level: In the impossible level, the developer implements a lockout mechanism to slow down attackers.

X. Command Injection

In the next steps, you will perform a command execution, sometimes called a command injection, attack on the DVWA. A command execution attack takes advantage of an application that allows the user to execute OS commands, such as Ping, via the Web server. Improperly secured applications could allow user to execute any command. In this case, you will use the DVWA to retrieve the contents for the users file.

1. Set Security Low
2. Go to Command Injection
3. Enter '127.0.0.1' and take snapshot



The screenshot shows the DVWA web application running on localhost. The browser address bar displays 'localhost/DVWA-master/vulnerabilities/exec/#'. The DVWA logo is at the top. A sidebar on the left contains navigation links: Home, Instructions, Setup / Reset DB, Brute Force, Command Injection (highlighted in green), CSRF, File Inclusion, File Upload, Insecure CAPTCHA, SQL Injection, SQL Injection (Blind), Weak Session IDs, XSS (DOM), XSS (Reflected), XSS (Stored), CSP Bypass, JavaScript, DVWA Security, PHP Info, and About. The main content area is titled 'Vulnerability: Command Injection'. It features a 'Ping a device' section with a text input field for 'Enter an IP address:' and a 'Submit' button. Below the input field, the output of a ping command to 127.0.0.1 is displayed in red text. At the bottom, a 'More Information' section lists three links: <https://www.scribd.com/doc/2530476/Php-Endangers-Remote-Code-Execution>, <http://www.ss64.com/bash/>, and <http://www.ss64.com/nt/>.

Vulnerability: Command Injection

Ping a device

Enter an IP address:

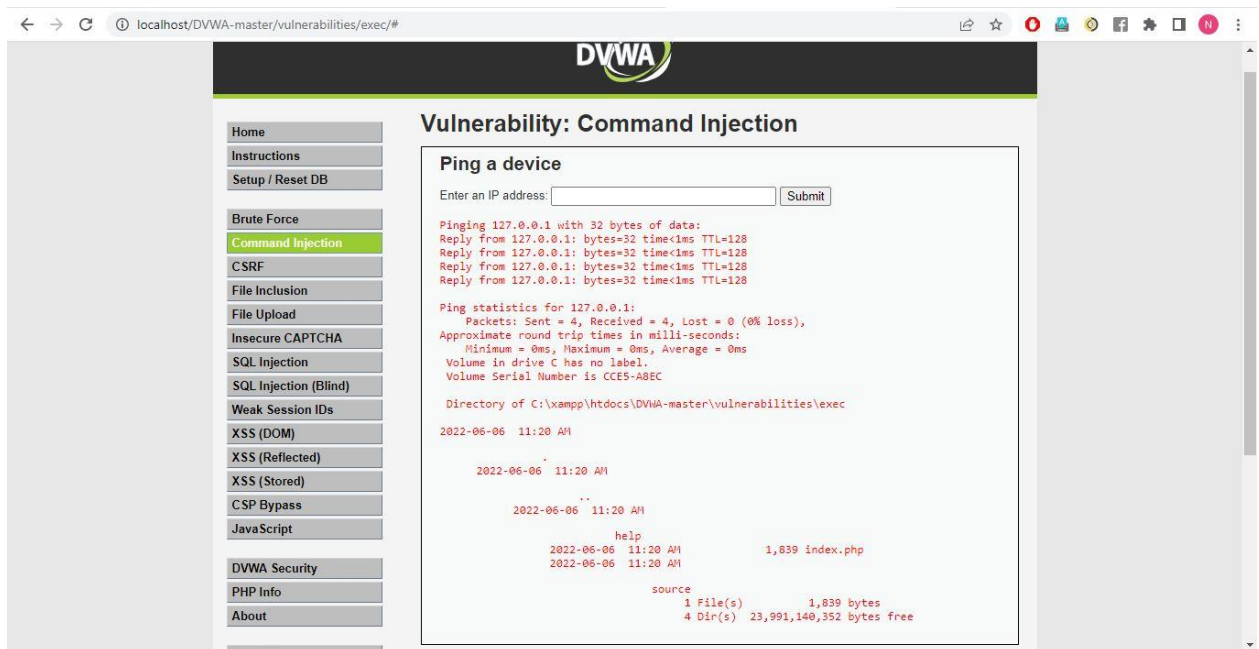
Pinging 127.0.0.1 with 32 bytes of data:
Reply from 127.0.0.1: bytes=32 time<1ms TTL=128
Reply from 127.0.0.1: bytes=32 time<1ms TTL=128
Reply from 127.0.0.1: bytes=32 time<1ms TTL=128
Reply from 127.0.0.1: bytes=32 time<1ms TTL=128

Ping statistics for 127.0.0.1:
Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
Minimum = 0ms, Maximum = 0ms, Average = 0ms

More Information

- <https://www.scribd.com/doc/2530476/Php-Endangers-Remote-Code-Execution>
- <http://www.ss64.com/bash/>
- <http://www.ss64.com/nt/>
- https://owasp.org/www-community/attacks/Command_Injection

4. Enter '127.0.0.1 && dir' and take snapshot



The screenshot shows the DVWA (Damn Vulnerable Web Application) interface. The left sidebar contains a menu with various security modules, and 'Command Injection' is highlighted. The main content area is titled 'Vulnerability: Command Injection' and features a 'Ping a device' section. Below the input field, the output of the command '127.0.0.1 && dir' is displayed in red text, showing the directory structure of the web application. The output includes the directory path 'C:\xampp\htdocs\DVWA-master\vulnerabilities\exec' and a list of files and directories, including 'index.php' and 'source'.

```
Enter an IP address:  Submit

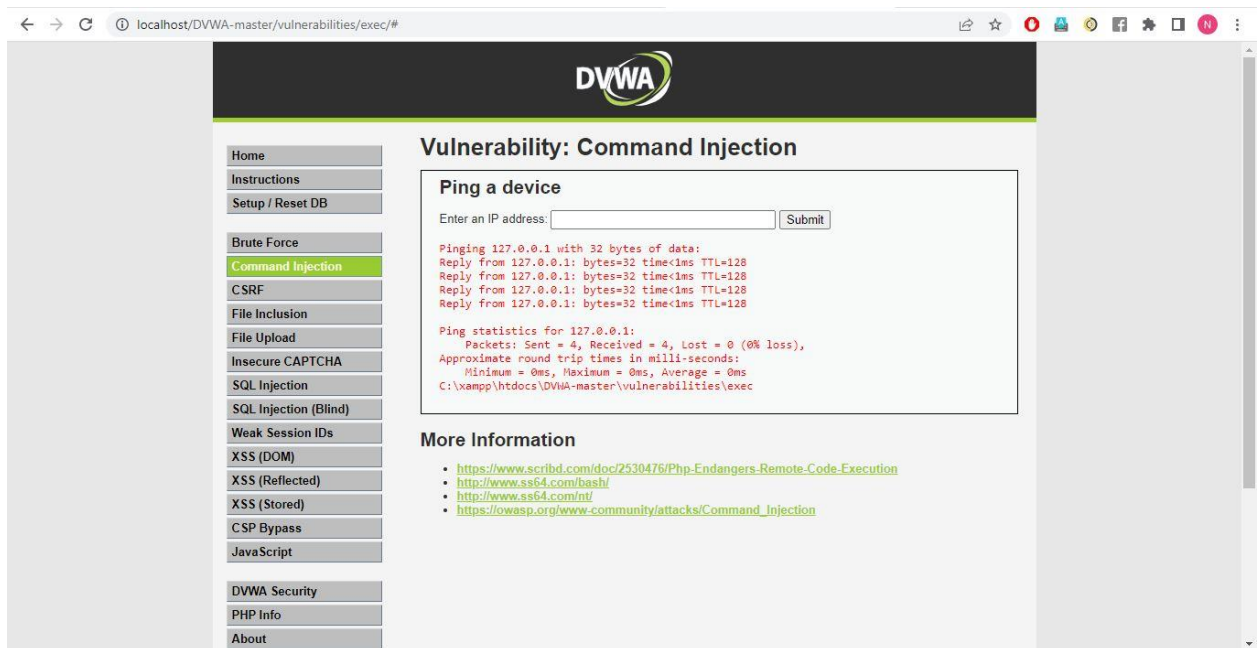
Pinging 127.0.0.1 with 32 bytes of data:
Reply from 127.0.0.1: bytes=32 time<1ms TTL=128
Reply from 127.0.0.1: bytes=32 time<1ms TTL=128
Reply from 127.0.0.1: bytes=32 time<1ms TTL=128
Reply from 127.0.0.1: bytes=32 time<1ms TTL=128
Reply from 127.0.0.1: bytes=32 time<1ms TTL=128

Ping statistics for 127.0.0.1:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
    Approximate round trip times in milli-seconds:
        Minimum = 0ms, Maximum = 0ms, Average = 0ms
    Volume in drive C has no label.
    Volume Serial Number is CCE5-A8EC

    Directory of C:\xampp\htdocs\DVWA-master\vulnerabilities\exec

2022-06-06  11:20 AM
                .
                ..
                2022-06-06  11:20 AM
                        help
2022-06-06  11:20 AM                  1,839 index.php
2022-06-06  11:20 AM
                        source
                        1 File(s)          1,839 bytes
                        4 Dir(s)  23,991,140,352 bytes free
```

5. Enter '127.0.0.1 && cd' and take snapshot



The screenshot shows the DVWA (Damn Vulnerable Web Application) interface. The left sidebar contains a menu with various security modules, and 'Command Injection' is highlighted. The main content area is titled 'Vulnerability: Command Injection' and features a 'Ping a device' section. Below the input field, the output of the command '127.0.0.1 && cd' is displayed in red text, showing the directory structure of the web application. The output includes the directory path 'C:\xampp\htdocs\DVWA-master\vulnerabilities\exec' and a list of files and directories, including 'index.php' and 'source'.

```
Enter an IP address:  Submit

Pinging 127.0.0.1 with 32 bytes of data:
Reply from 127.0.0.1: bytes=32 time<1ms TTL=128
Reply from 127.0.0.1: bytes=32 time<1ms TTL=128
Reply from 127.0.0.1: bytes=32 time<1ms TTL=128
Reply from 127.0.0.1: bytes=32 time<1ms TTL=128
Reply from 127.0.0.1: bytes=32 time<1ms TTL=128

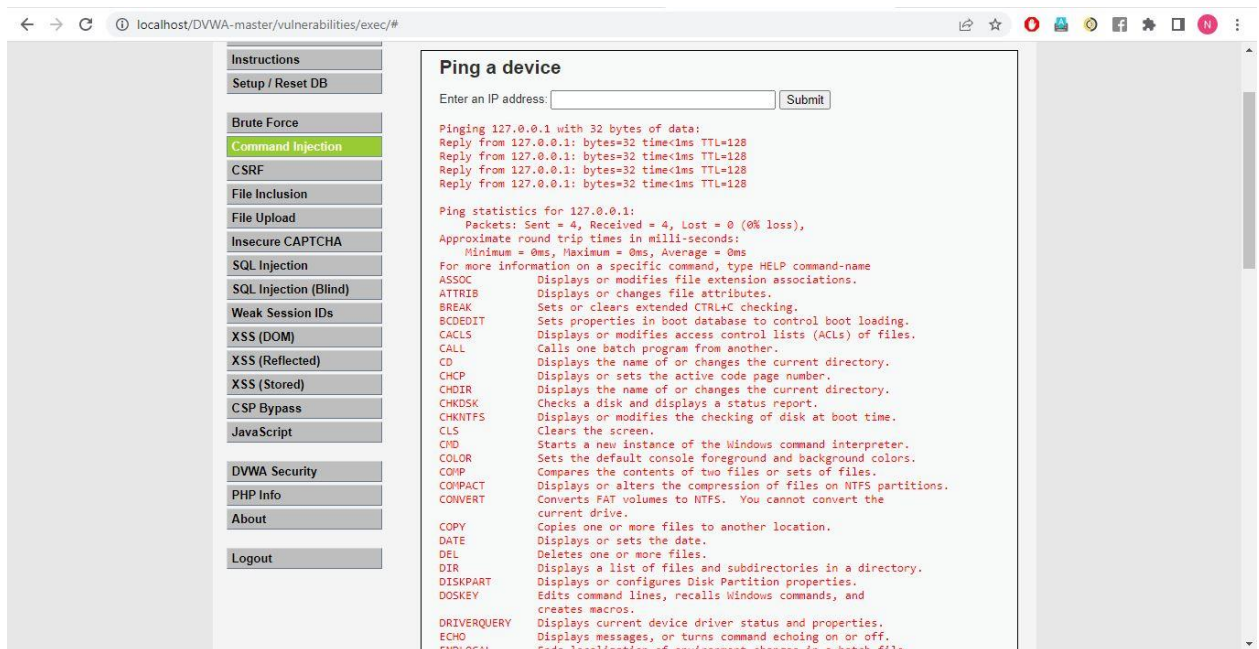
Ping statistics for 127.0.0.1:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
    Approximate round trip times in milli-seconds:
        Minimum = 0ms, Maximum = 0ms, Average = 0ms
    C:\xampp\htdocs\DVWA-master\vulnerabilities\exec

More Information

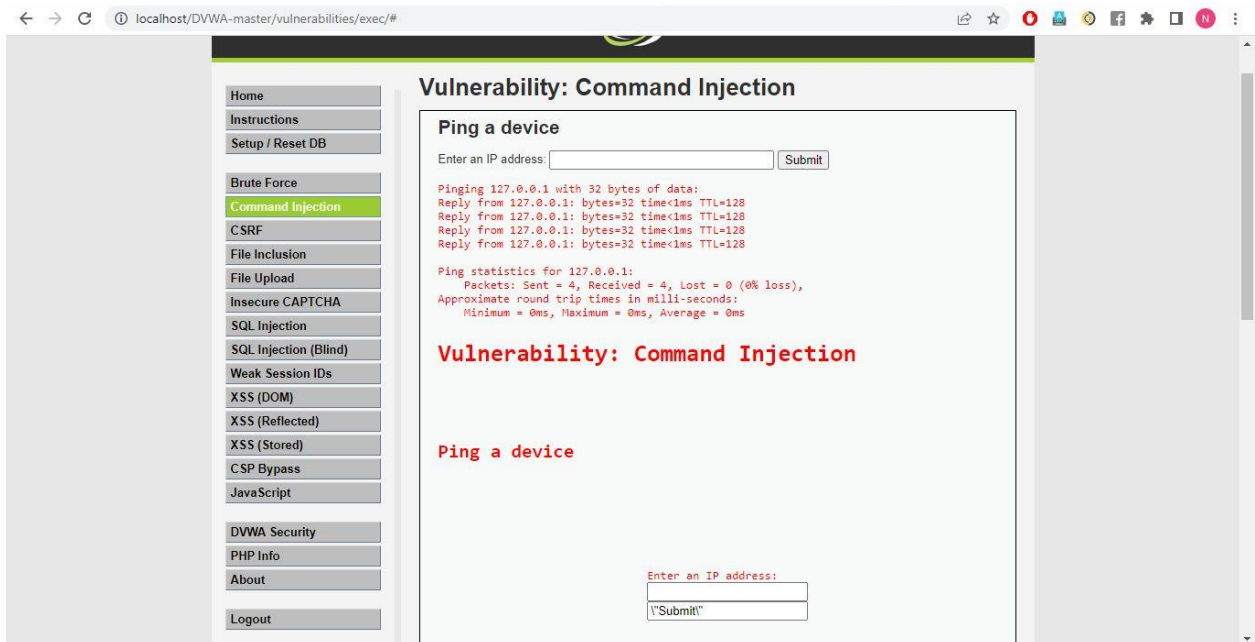

- https://www.scribd.com/doc/2530476/Php-Endangers-Remote-Code-Execution
- http://www.ss64.com/bash/
- http://www.ss64.com/ht/
- https://owasp.org/www-community/attacks/Command\_injection

```


6. Enter '127.0.0.1 && help' and take snapshot



7. Enter '127.0.0.1 && type index.php' and take snapshot



About

Logout

Enter an IP address:

Submit

```
\n";
if( $vulnerabilityFile == 'impossible.php' )
    $page[ 'body' ] .= "          " . tokenField();
$page[ 'body' ] .= "

        { $html }

More Information

    • " . dvwaExternalLinkUrlGet( 'https://www.scribd.com/doc/2530476/Php-Endangers-Remote-Code-Execution' ) . "
    • " . dvwaExternalLinkUrlGet( 'http://www.ss64.com/bash/' ) . "
    • " . dvwaExternalLinkUrlGet( 'http://www.ss64.com/nt/' ) . "
    • " . dvwaExternalLinkUrlGet( 'https://owasp.org/www-community/attacks/Command_Injection' ) . "

\n"; dvwaHtmlEcho( $page ); ?>
```

More Information