U.S. DEPARTMENT OF ENERGY'S
CYBERFORCE® PROGRAM

CyberForce® 101

Command Injection

October 2023

cyberforcecompetition@anl.gov

Introduction to Command Injections

Command Injection

A command line injection is used by attackers to execute commands on a host operating system through a vulnerable application. It is similar to a SQL injection attack in that it aims to execute commands to get more access to a system than other users.

A normal interaction between a user and server goes through an application. The user sends a message to the application which sends it to the server. Then, the server collects the information needed and sends it to the application. Sometimes the application sends that information to the user, but it can also store it for further use. If the proper precautions are not taken, an attacker can add an unauthorized command to the message sent to the application and gain too much access to the server and its contents.

Attack Process

Let's say we have a https website and we want to see if it is susceptible to an injection attack. First, we look at the website itself and see that the link has variable names and values. These will be used by the website to send commands. The fake website link is:

https://realy-bad-website.com/itemStatus?itemID=1234&itemNum=45

From this, we can see these variables and values:

itemID= 1234

itemNum= 45

A command the application could send with these values is:

itemReport.p1 1234 45

Second, we craft a command that will return a value if it is executed. This way we know if the application is vulnerable to this kind of

attack. Echo is a commonly used command for this. The command we will inject is:

```
& echo aiwefwlguh &
```

Third, we put the injection command as input for a command the system will be executing. This means that we will replace the itemID with our injection command.

```
stockreport.p1 & echo aiwefwlguh & 45
```

The character & is used to separate commands. As a result, the system reads this as three separate commands (stockreport.p1, echo aiwefwlguh and echo aiwefwlguh). This means that if the injection command is executed it will result in an error. This error may look like this:

```
Error itemID not found
```

After we have successfully tested the system, we can inject almost any command into it.

Blind Attack

Often times when a user interacts with a server through an application, they do not receive feedback. As a result, we need to execute a blind attack. This means that we don't automatically get the results of our commands. For example, a service such as leaving a comment on a website does not give the user any feedback. The server takes the comment and sends it to the admin, so there is no need to contact the user. The system can use the mail command:

```
mail -s "This is my comment" -aFrom:abc123-user.net comment@website-website.com
```

The application does not receive any output, so the standard echo command we used earlier will not work. A popular technique is to use the ping command to inject a timed pause. This is helpful because if the pause occurs, we know that the system is susceptible to injection attacks. The command for this is:

```
& ping -c 10 127.0.0.1 &
```

If this is successful, we can direct the output from the server to the web root which can be retrieved with our browser. For example, we can direct it to `/var/www/static/` with the command:

```
& whomi > /var/www/static/whoami.txt &
```

The `>` symbol sends the output from the `whoami` command to the designated file. This can be retrieved by opening the website in a browser with `/whoami.txt` at the end of the URL.

Sources

- https://portswigger.net/web-security/os-command-injection#:~:text=OS%20command%20injection%20(also%20known,application%20and%20all%20its%20data
- https://medium.com/@Steiner254/os-command-injection-c4ee28fab521
- https://youtu.be/UBWMLFbjPBc
    - (Skip to 2:24 to avoid advertisements/sponsors)