
Table of Contents

Download Links

Introduction	1.1
Virtual Machine (bsides-workshop-vm.ova)	1.2
Burp Suite Professional Edition	1.3
Burp Suite Community Edition	1.4

Initial Setup

Start Burp Suite Professional	2.1
Import Virtual Machines	2.2
Create New Firefox Profile	2.3
Add FoxyProxy Addon	2.4
Add New Proxy In FoxyProxy	2.5
Configure Proxy Listener	2.6
Install Burp's CA Certificate In Firefox	2.7
Access Security Shepherd Application	2.8
Remove Unnecessary Browser Traffic	2.9
Setup Hotkeys	2.10

Tools of the Trade

Target	3.1
Proxy	3.2
Scanner	3.3
Intruder	3.4
Repeater	3.5
Sequencer	3.6
Decoder	3.7
Comparer	3.8
Extender	3.9

Miscellaneous

About Us	4.1
----------	-----

Attacking Web Applications using Burp Suite

26th October, 2018

Workshop at:
[BSIDES DELHI 2018](#)

Abstract

In this completely hands-on workshop, you would get to understand the techniques and methodologies that could be applied when performing a web application penetration testing. Throughout this workshop, you would be using Burp Suite tool, which is a conglomerate of distinct tools with powerful features. Apart from gaining familiarity with the tools and the techniques involved in application security testing, you would also get an opportunity to understand some of the common vulnerabilities from the OWASP Top 10 – 2017 list. We would provide you with a vulnerable website, and you would uncover security issues in it even if you h# Attacking Web Applications using Burp Suite

26th October, 2018

Workshop at:
[BSIDES DELHI 2018](#)

Abstract

In this completely hands-on workshop, you would get to understand the techniques and methodologies that could be applied when performing a web application penetration testing. Throughout this workshop, you would be using Burp Suite tool, which is a conglomerate of distinct tools with powerful features. Apart from gaining familiarity with the tools and the techniques involved in application security testing, you would also get an opportunity to understand some of the common vulnerabilities from the OWASP Top 10 – 2017 list. We would provide you with a vulnerable website, and you would uncover security issues in it even if you have never done this before!

Speaker Profile

Speaker-1

Riddhi Shree is working as Application Security Engineer at Appsecco. She is also one of the chapter leaders for null Community - Bangalore Chapter. She has over 9 years of experience in Software testing industry. She has delivered multiple talks and training sessions during open security meetups. She has, recently, conducted a 2-days workshop at c0c0n XI - Data Privacy, Cyber Security & Hacking Conference on "Burp Suite For Web and Mobile Security Testing".

Speaker-2

Vandana Verma is a Security Architect at IBM with over 12 years of experience specializing in web application, infrastructure and cloud security. She is part of security communities such as Volunteer Coordinator – Asia Pacific for OWASP Women in Appsec (WIA) & OWASP WIA Secretary, OWASP Chapter Leader and Heading InfoSecGirls. She has given talks and workshops at many colleges and security conferences including AppSec Europe, AppSec USA, NullCon and c0c0n.

Pre-Requisites

- Laptop with administrator access (mandatory)
- Minimum 4 GB RAM
- Atleast 10 GB of free hard disk space
- Oracle VirtualBox 5.x or later installed.
- Burp Suite Professional / Community Edition installed
- Make sure Burp Suite can start
- Firefox browser with FoxyProxy Standard add-on installed

Download Links

- [Virtual Machine \(bsides-workshop-vm.ova\)](#)
- [Burp Suite Professional Edition](#)
- [Burp Suite Community Edition](#)
- [FoxyProxy Standard add-on](#) ave never done this before!

Speaker Profile

Speaker-1

Riddhi Shree is working as Application Security Engineer at Appsecco. She is also one of the chapter leaders for null Community - Bangalore Chapter. She has over 9 years of experience in Software testing industry. She has delivered multiple talks and training sessions during open security meetups. She has, recently, conducted a 2-days workshop at c0c0n XI - Data Privacy, Cyber Security & Hacking Conference on "Burp Suite For Web and Mobile Security Testing".

Speaker-2

Vandana Verma is a Security Architect with over 12 years of experience specializing in web application, infrastructure and cloud security. She is part of security communities such as Volunteer Coordinator – Asia Pacific for OWASP Women in Appsec (WIA) & OWASP WIA Secretary, OWASP Chapter Leader and Heading InfoSecGirls. She has given talks and workshops at many colleges and security conferences including AppSec Europe, AppSec USA, NullCon and c0c0n.

Pre-Requisites

1. Download files from the provided download links (see section below).
2. Make sure you have done the initial setup by following the steps as described in the **Initial Setup** section.
3. Familiarize yourself with the Burp tools by following the steps mentioned in **Tools of the Trade** section.

Download Links

- [Virtual Machine \(bsides-workshop-vm.ova\)](#)
- [Burp Suite Professional Edition](#)
- [Burp Suite Community Edition](#)

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](#)

Start Burp Suite Professional

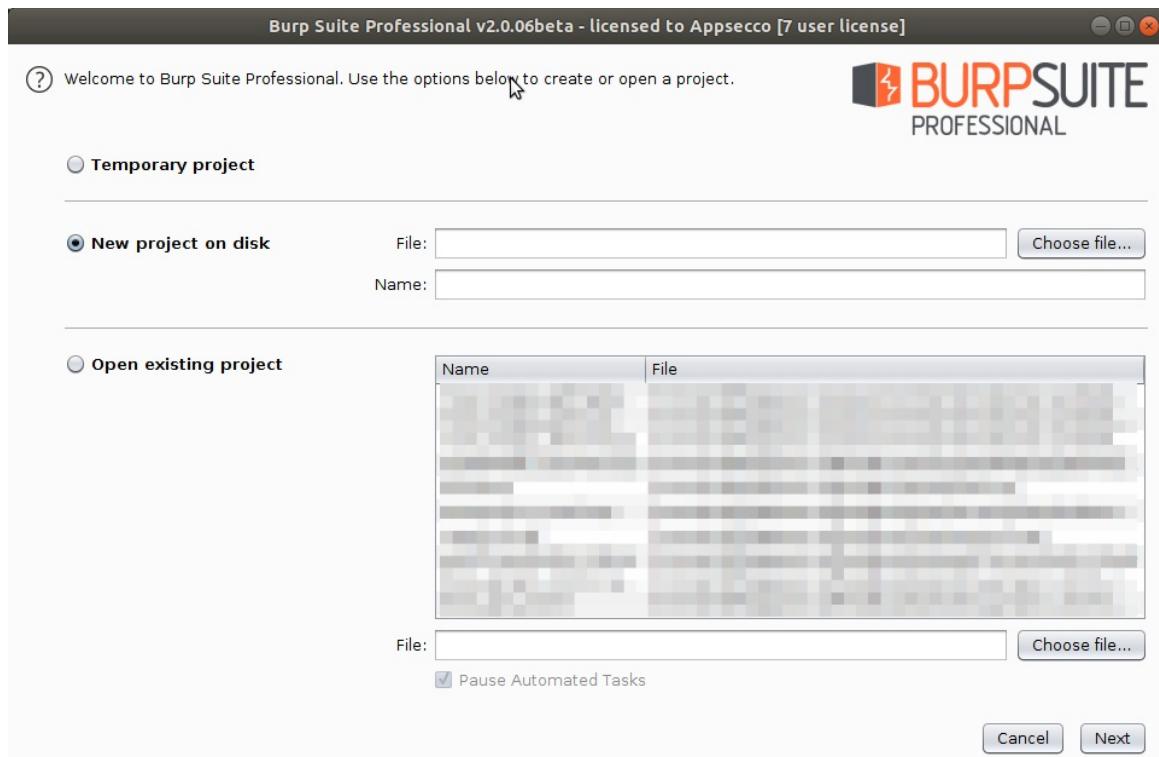
1. Download the JAR file for [Burp Suite Professional](#).
2. Go to the folder containing the jar file "burpsuite_pro_v2.0.07beta.jar".



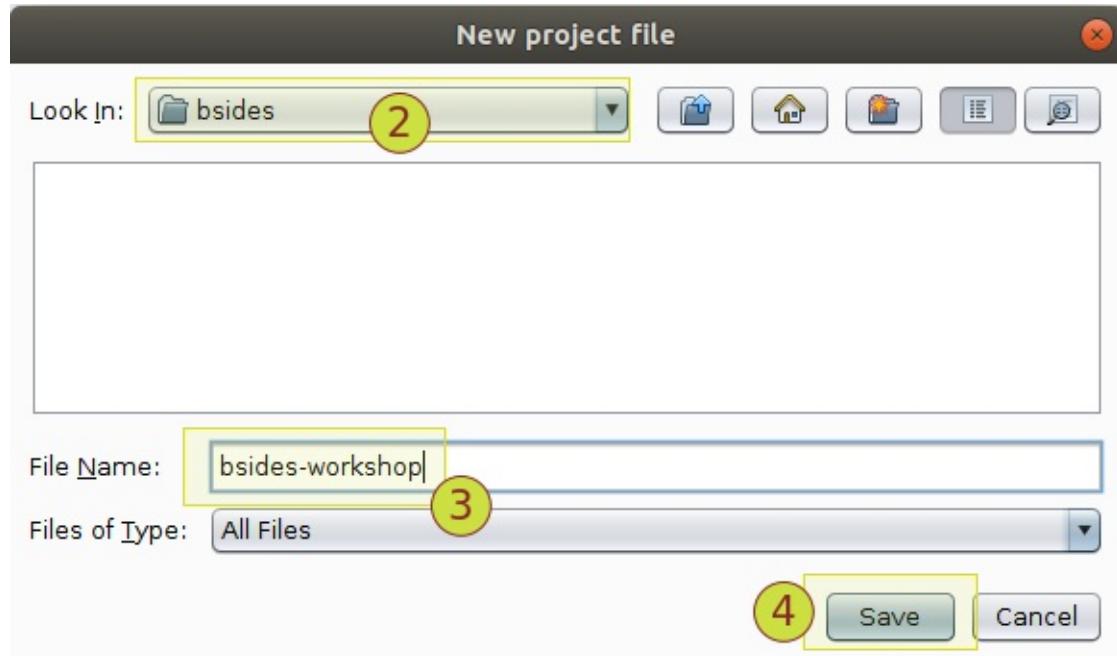
3. Run following command to start Burp Suite Professional Edition.

```
$ java -jar -Xmx2G burpsuite_pro_v2.0.07beta.jar
```

4. If you are prompted for licence, enter licence.
5. If you do not have a licence, switch to [Burp Suite Community Edition](#), and skip to step #7.
6. Select "New project on disk" radio button.



7. Click on "Choose file" button, and choose a new location for creating a new project.



8. Click on "Next".
9. Click on "Start Burp".
10. Go to "Proxy" > "Options" > "Proxy Listeners", and validate the settings.

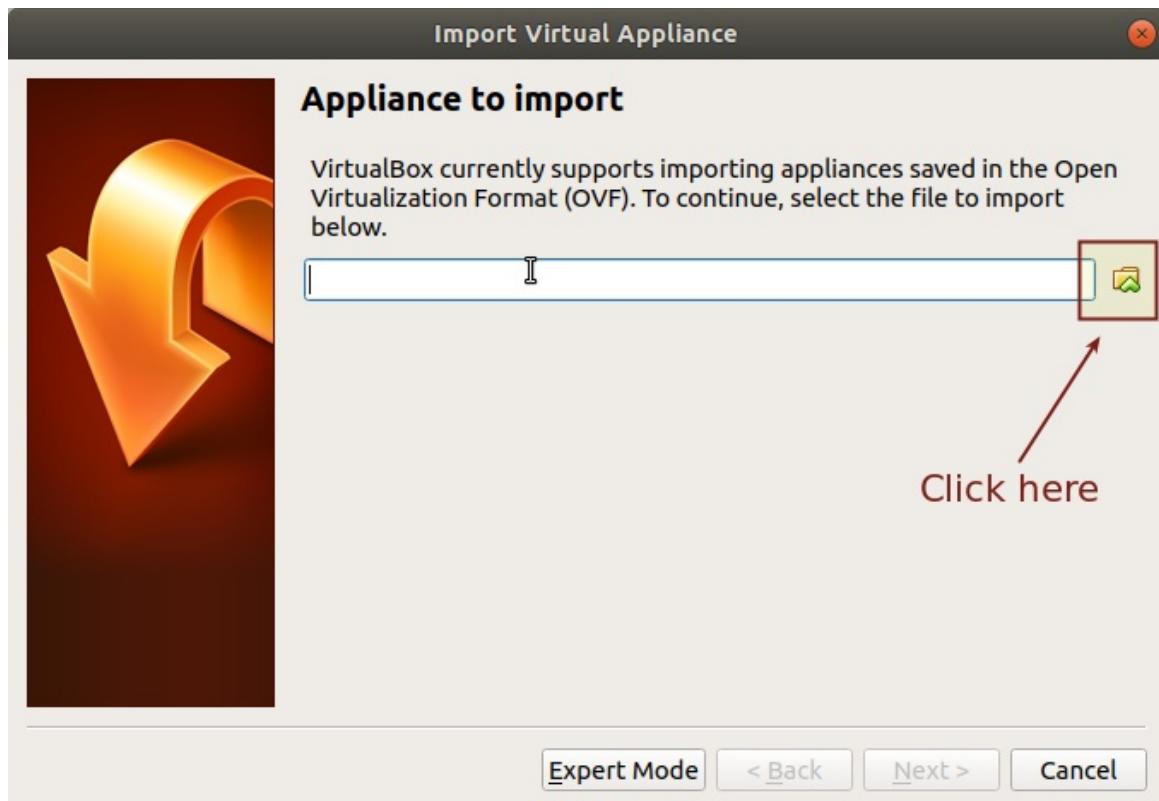
Running	Interface	Invisible	Redirect	Certificate
<input checked="" type="checkbox"/>	127.0.0.1:8080			Per-host

Import Virtual Machines

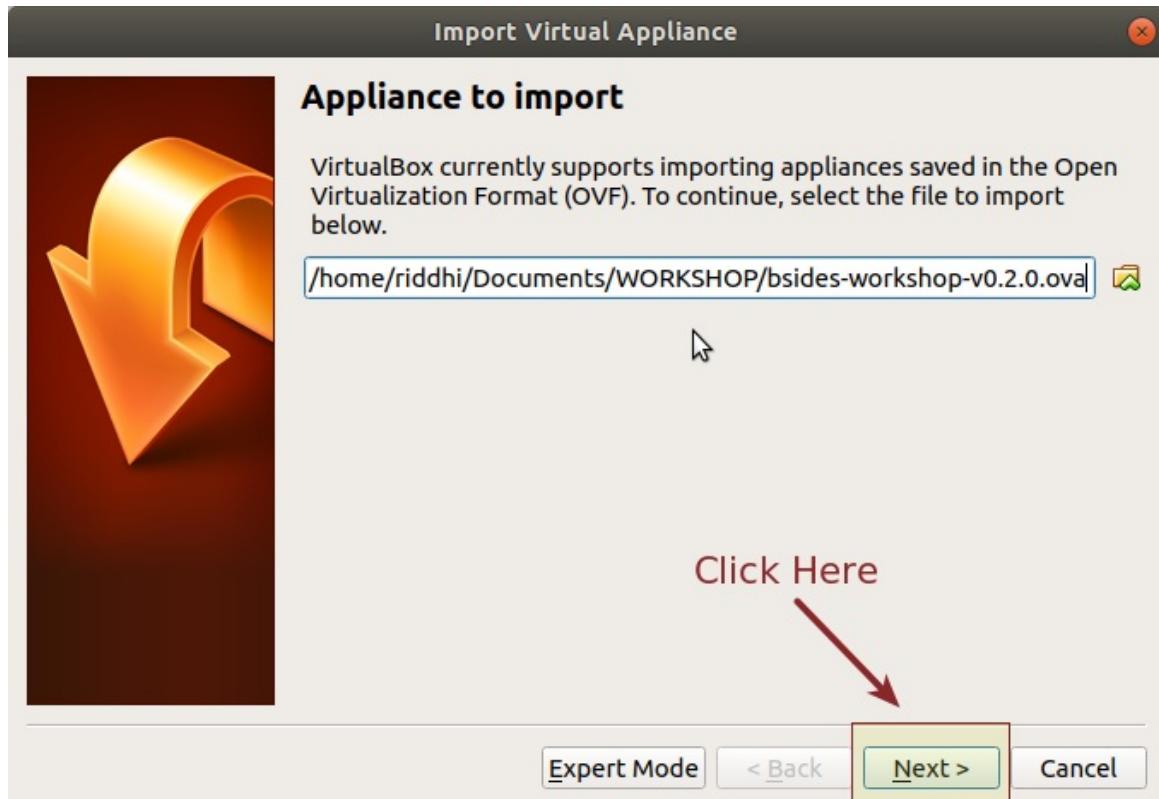
1. Download the OVA file named as [bsides-workshop-vm.ova](#).
2. Open Virtual Box.
3. Click on "File" > "Import Appliance"



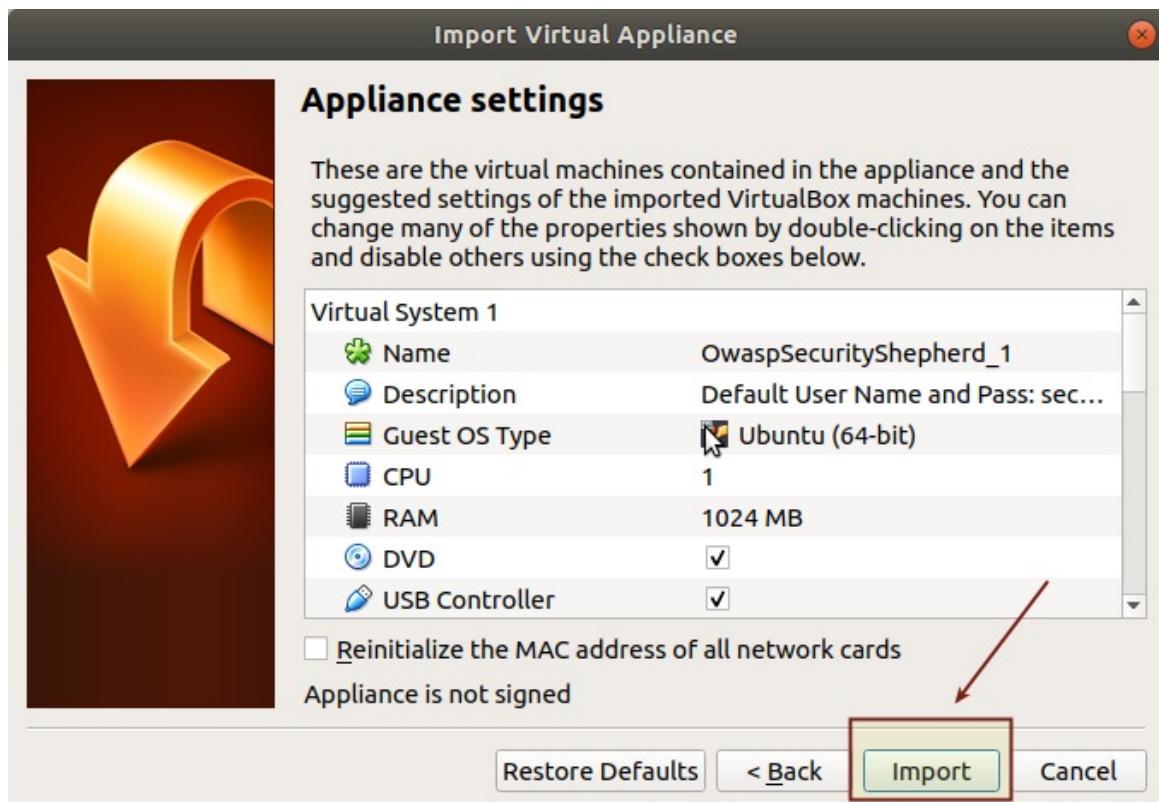
4. Locate and select the downloaded OVA file, i.e., "bsides-workshop-vm.ova".

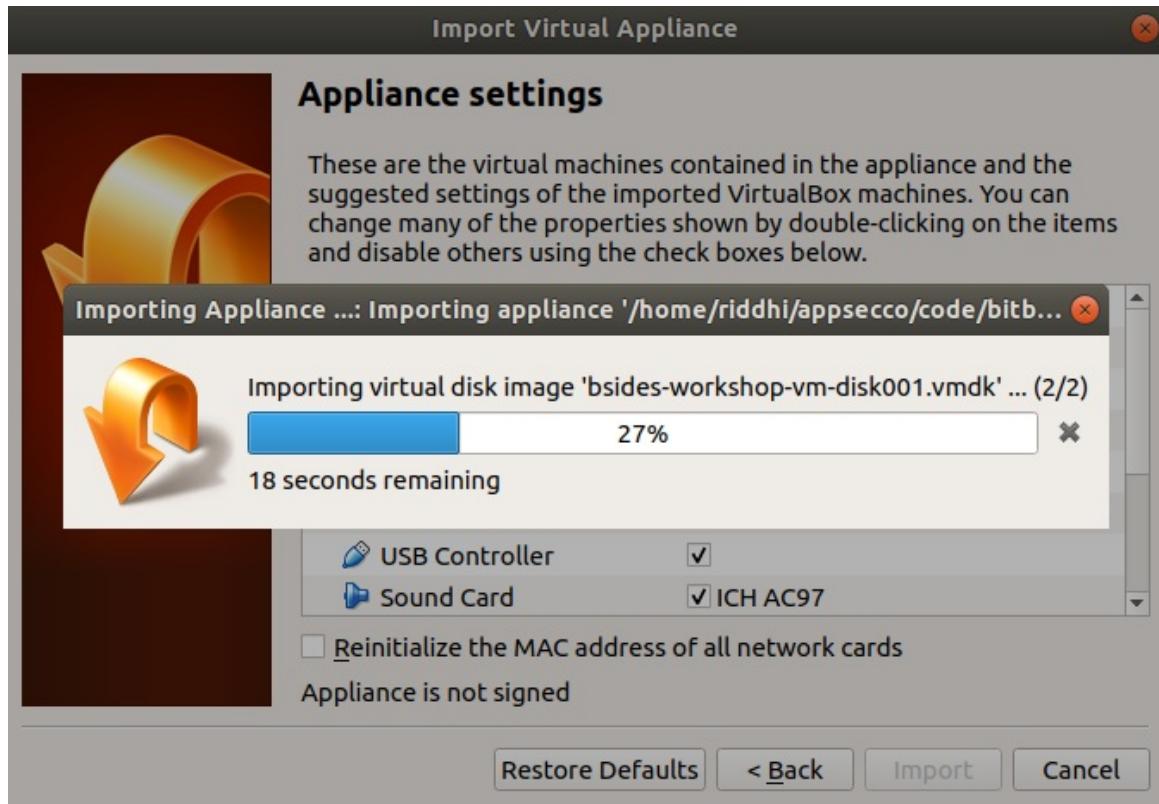


5. Click on "Next" button.



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





7. You should see a virtual machine named as "bsides-workshop" imported successfully.



Create a new Firefox Profile

1. Close all running instances of Firefox browser.
2. Open a terminal in Ubuntu/Mac.
3. In Windows, press "Windows + R" to open the Run dialog.
4. Run the following command:

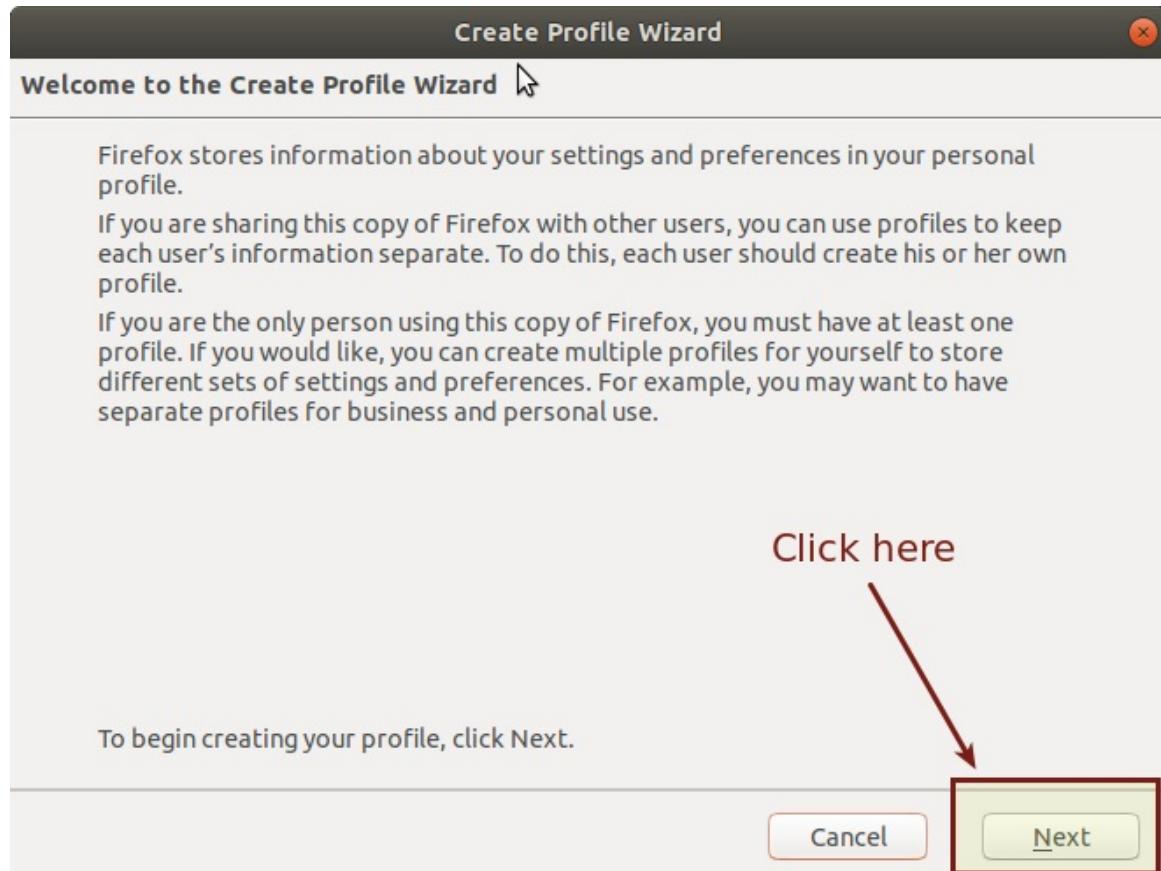
Linux/MacOS: `firefox -p / firefox -P / firefox -ProfileManager`

Windows:

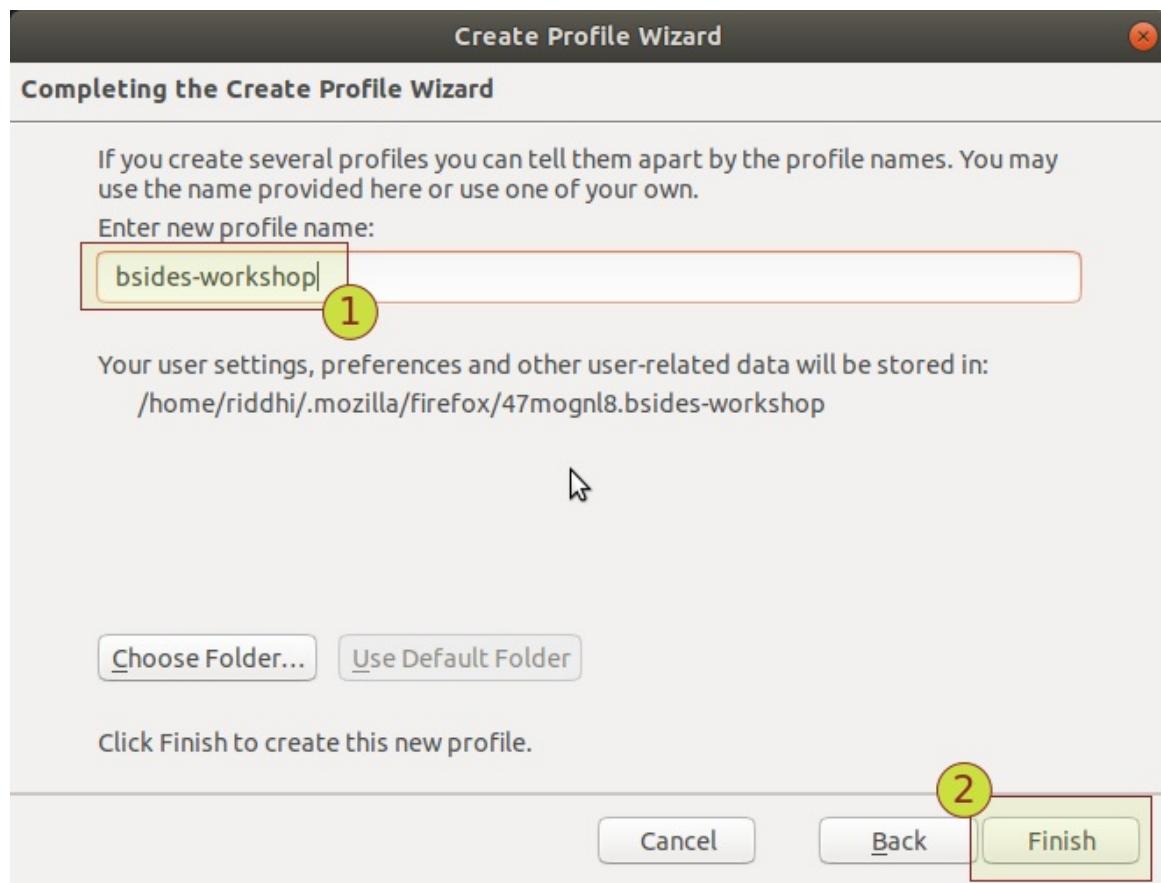
`firefox.exe -P`

5. Click on "Create Profile".





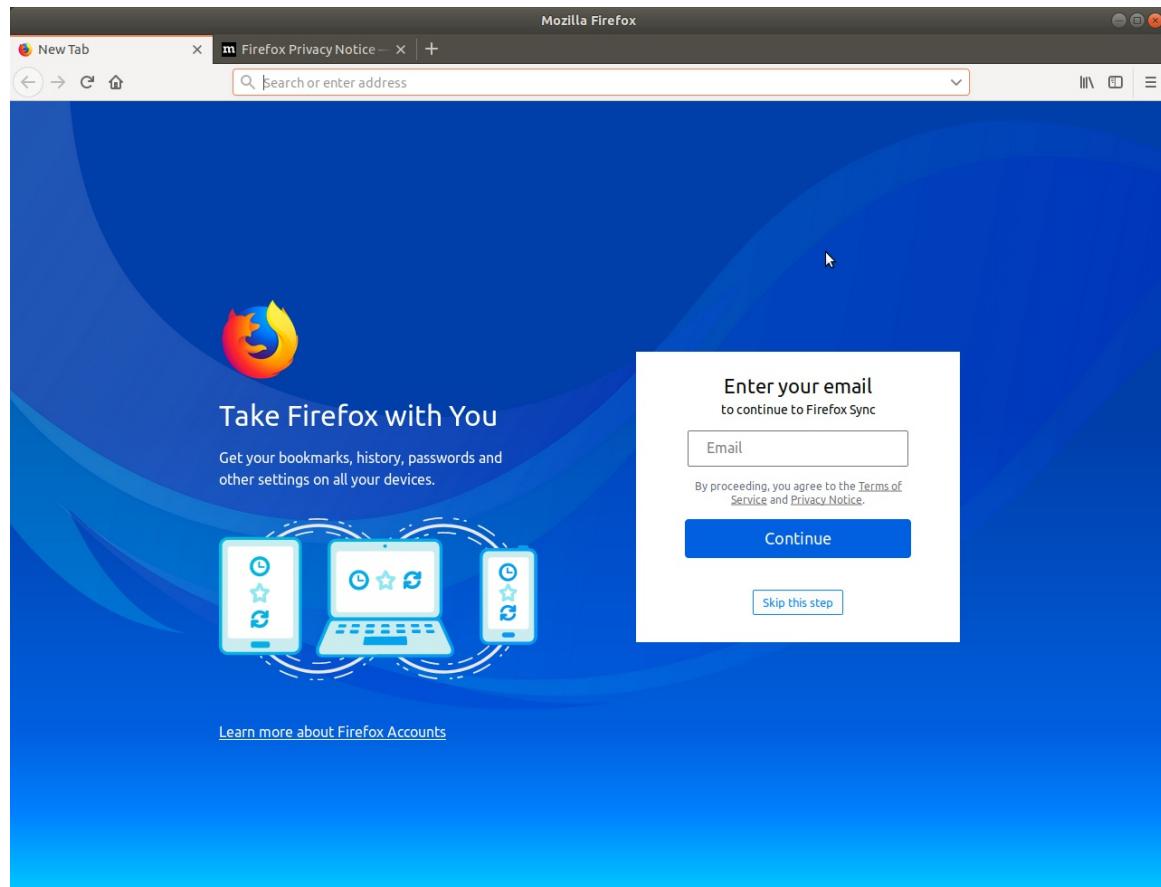
6. Give a name to your new firefox profile and click on "Finish" button.



7. Select the newly created profile and start firefox.

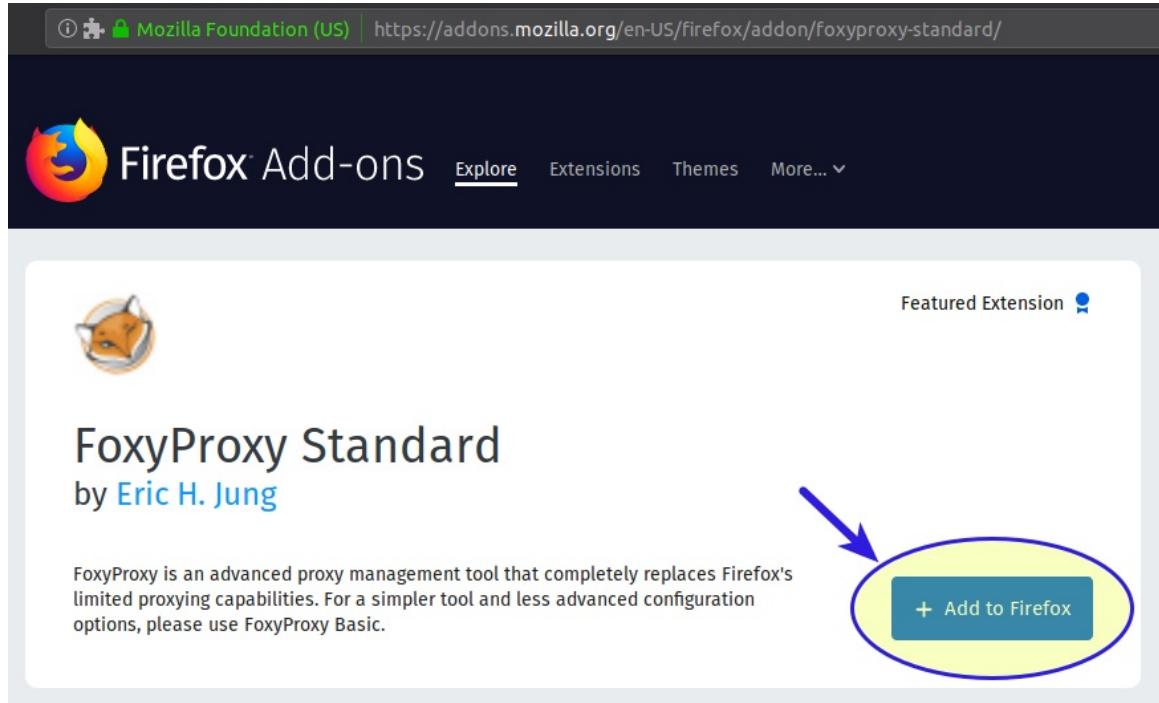


8. A fresh instance of Firefox browser should start with default browser settings.

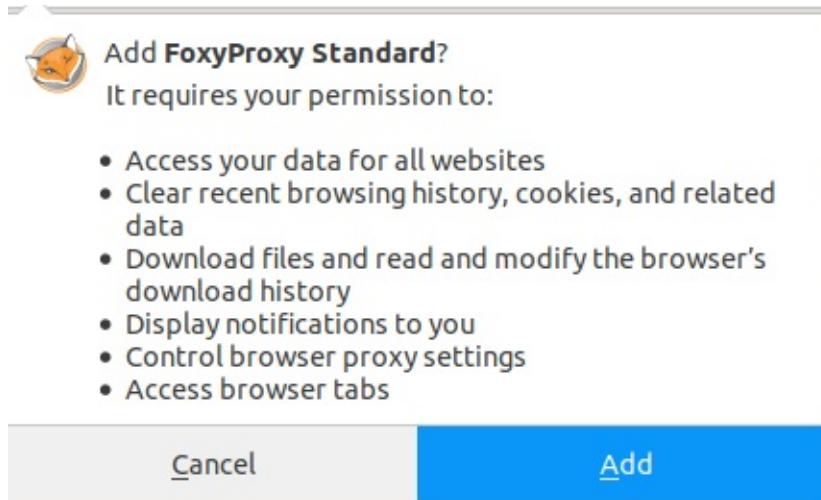


Add Foxy-Proxy Standard Add-on

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.



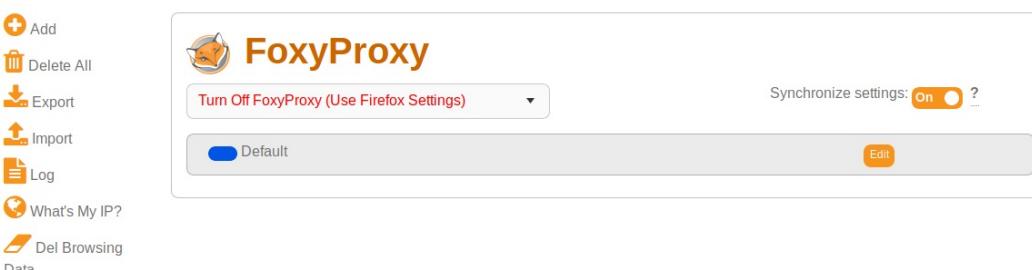
-- Eric H. Jung, May, 2018

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



FoxyProxy

Turn Off FoxyProxy (Use Firefox Settings)

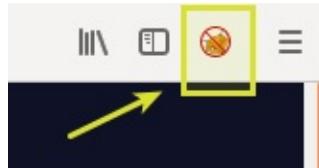
Synchronize settings: **On** ?

Default

Add **Delete All** **Export** **Import** **Log** **What's My IP?** **Del Browsing Data** **About**

Add a New Proxy in FoxyProxy

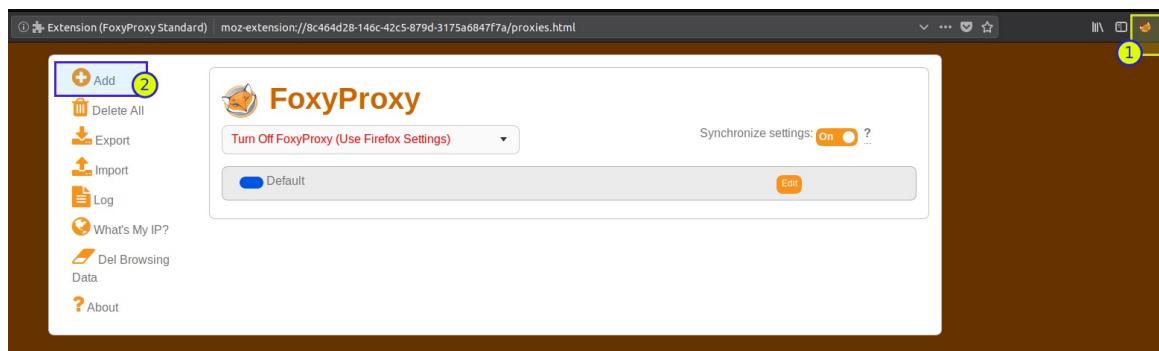
1. Go to the Firefox browser.
2. Click on `FoxyProxy` icon in the top-right corner of the browser.



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)

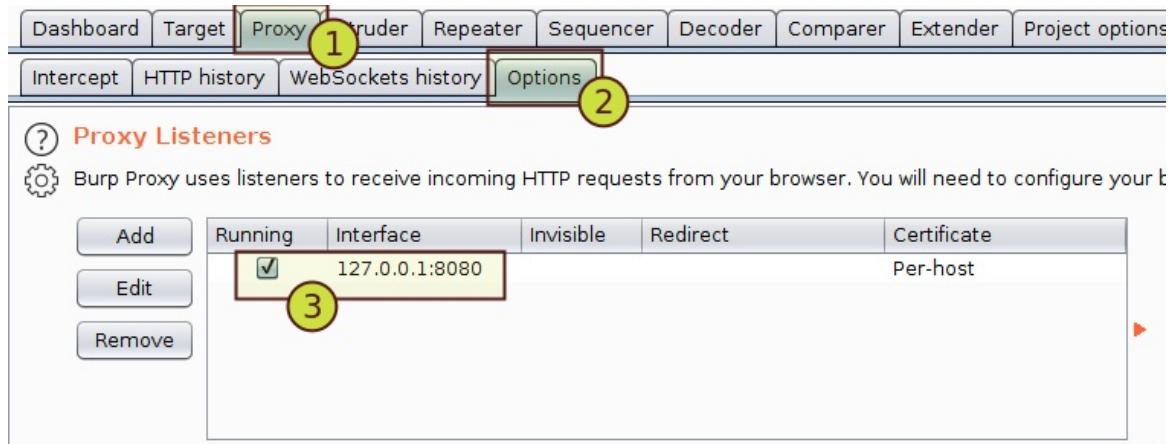
Synchronize settings: ?

localhost-8080 127.0.0.1

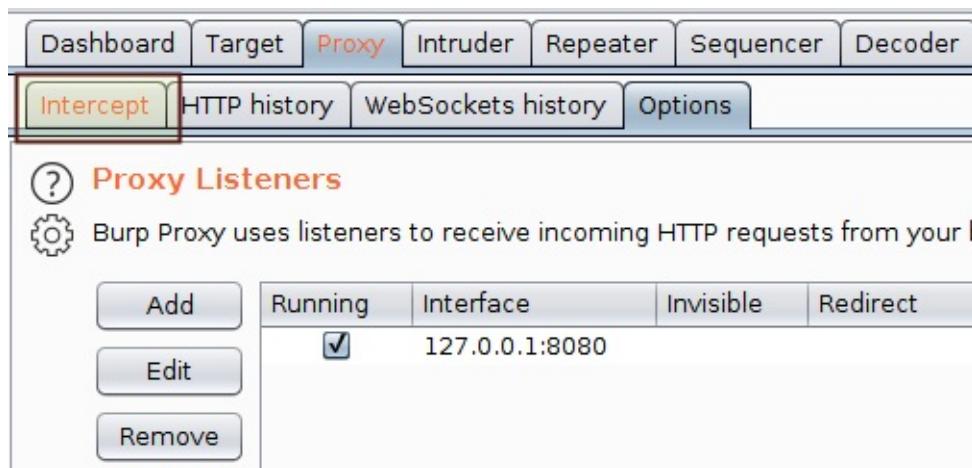
Default

Configure Proxy Listener in Burp Suite

- In Burp Suite, 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 .

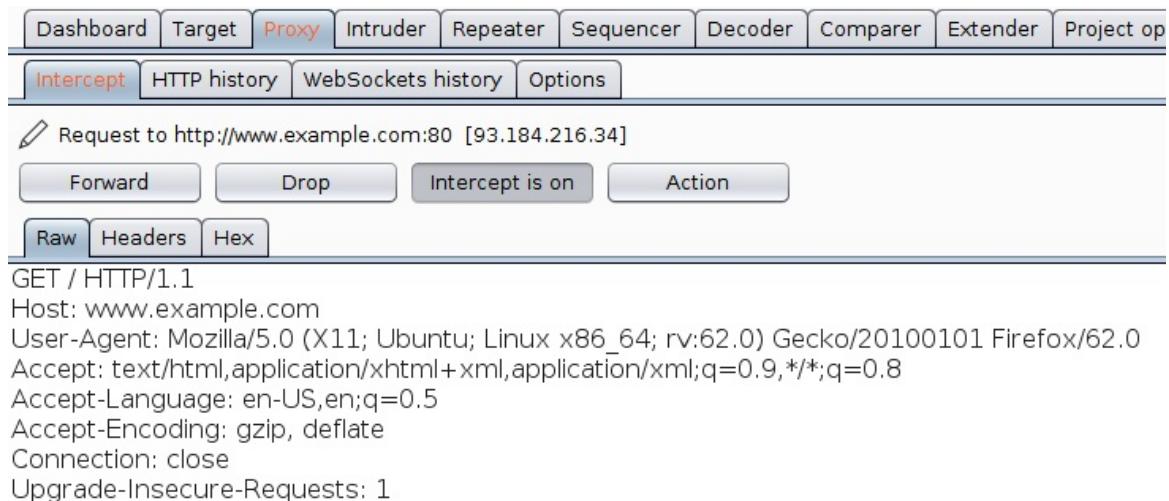


- Switch to your Firefox browser and navigate to `www.example.com`



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

- Forward the intercepted request.



5. Switch to "Proxy" > "HTTP History" tab.

#	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

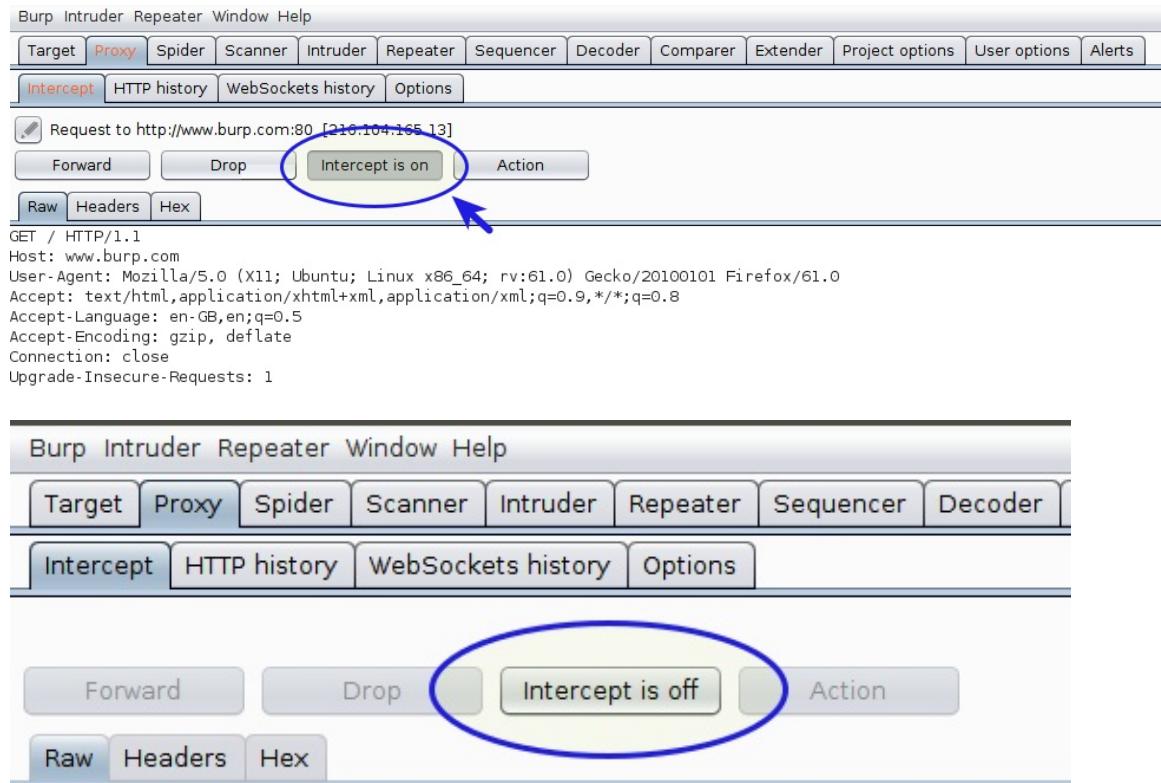
Raw Headers Hex HTML Render

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

6. You should see the traffic from Firefox browser getting populated in Burp's HTTP history table.

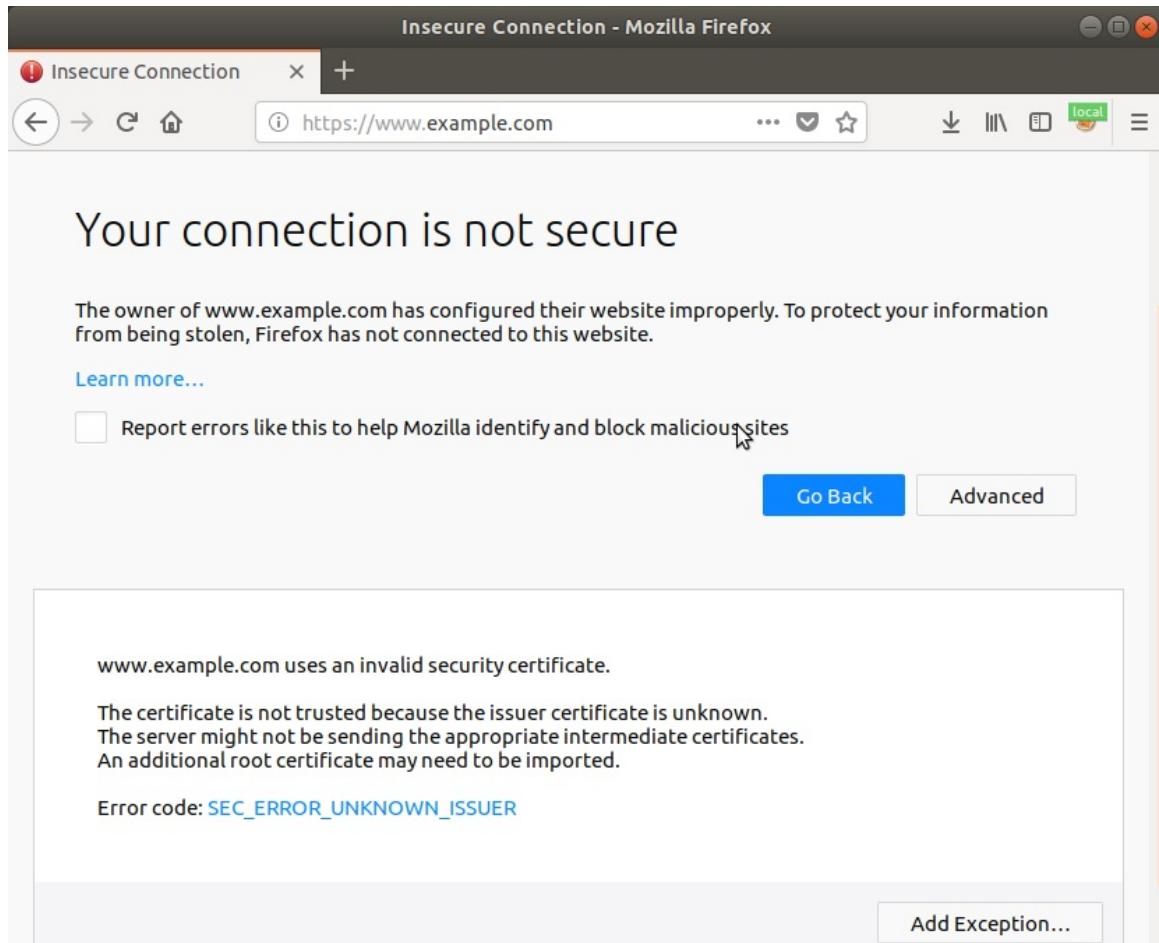
Install Burp's CA Certificate

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

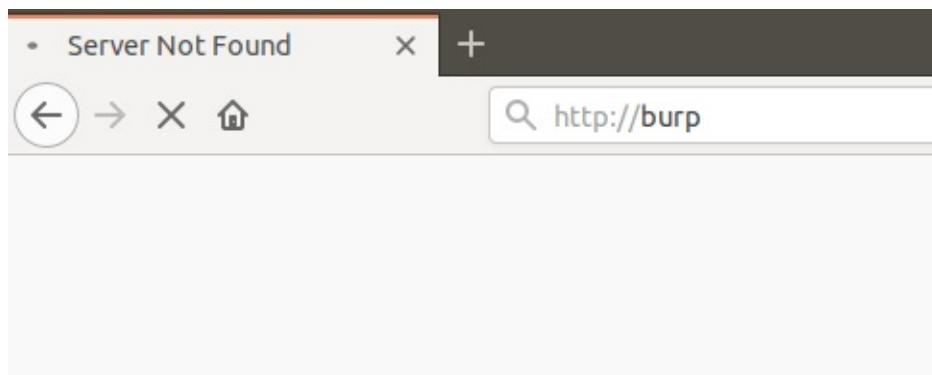


2. In Firefox, navigate to a secure website, e.g., <https://www.example.com>.

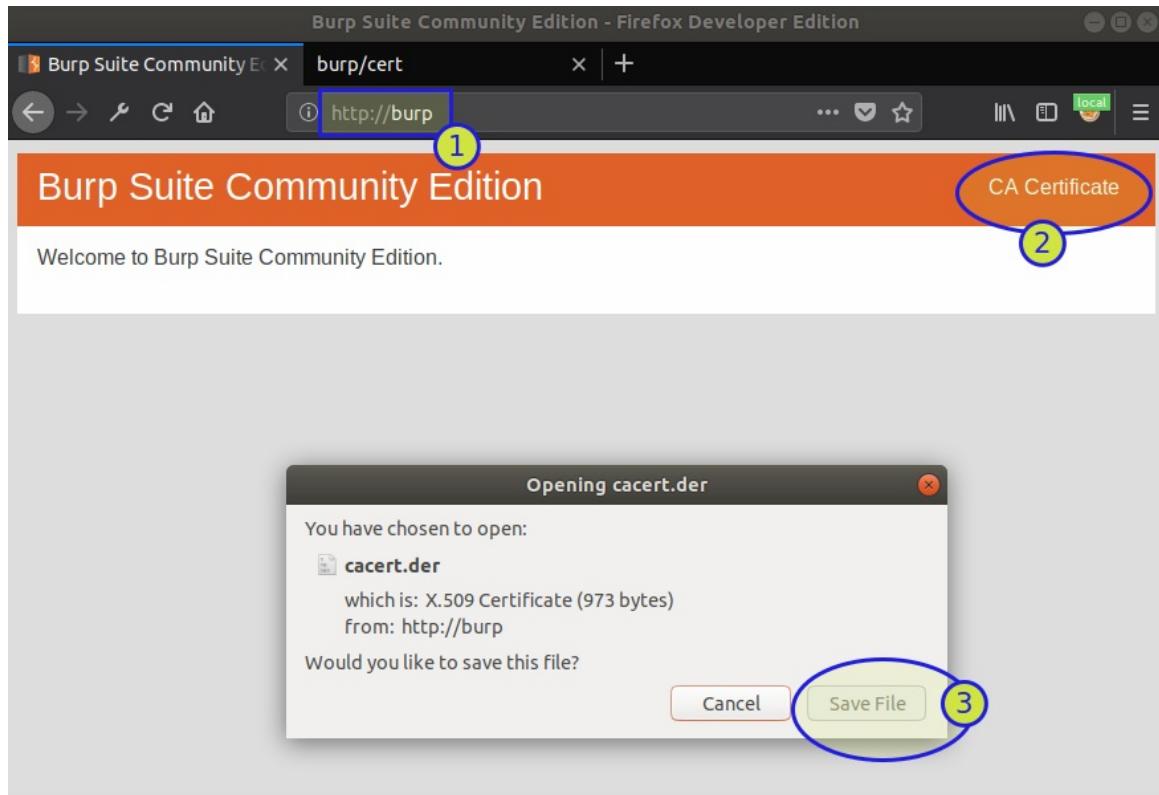
If you have configured Burp's proxy listener correctly, and you haven't installed Burp's self-signed Certificate Authority (CA) certificate, yet, then the browser may throw an "invalid security certificate" error with the message "...issuer certificate is unknown". Click on the "Advanced" button to see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



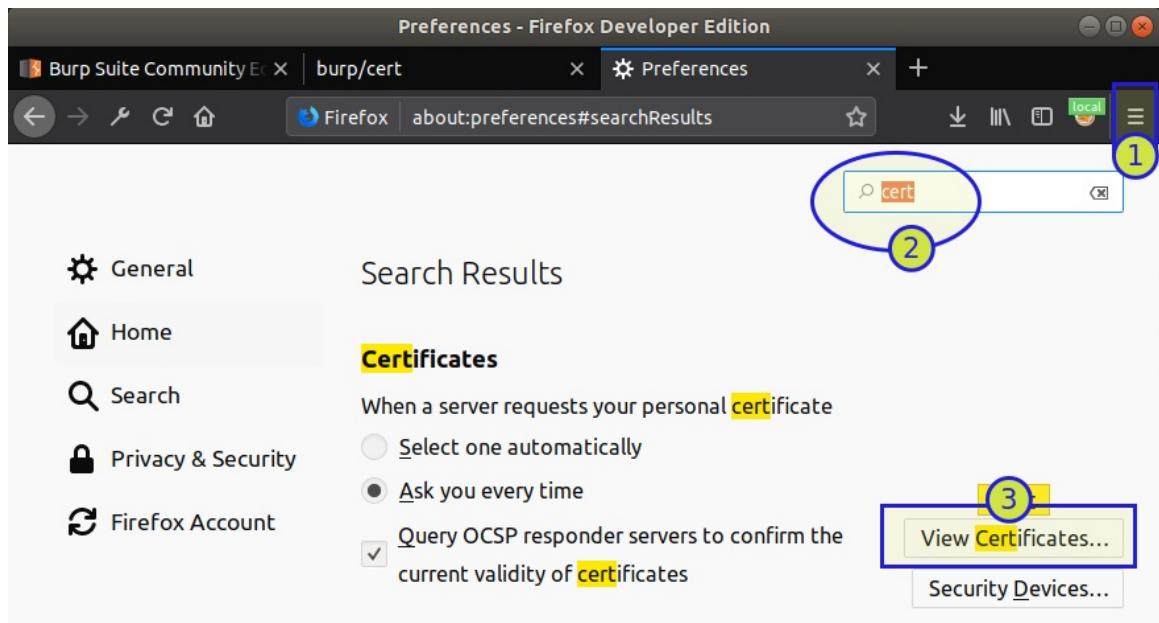
3. If you received "SEC_ERROR_UNKNOWN_ISSUER" error from the browser, navigate to `http://burp`.



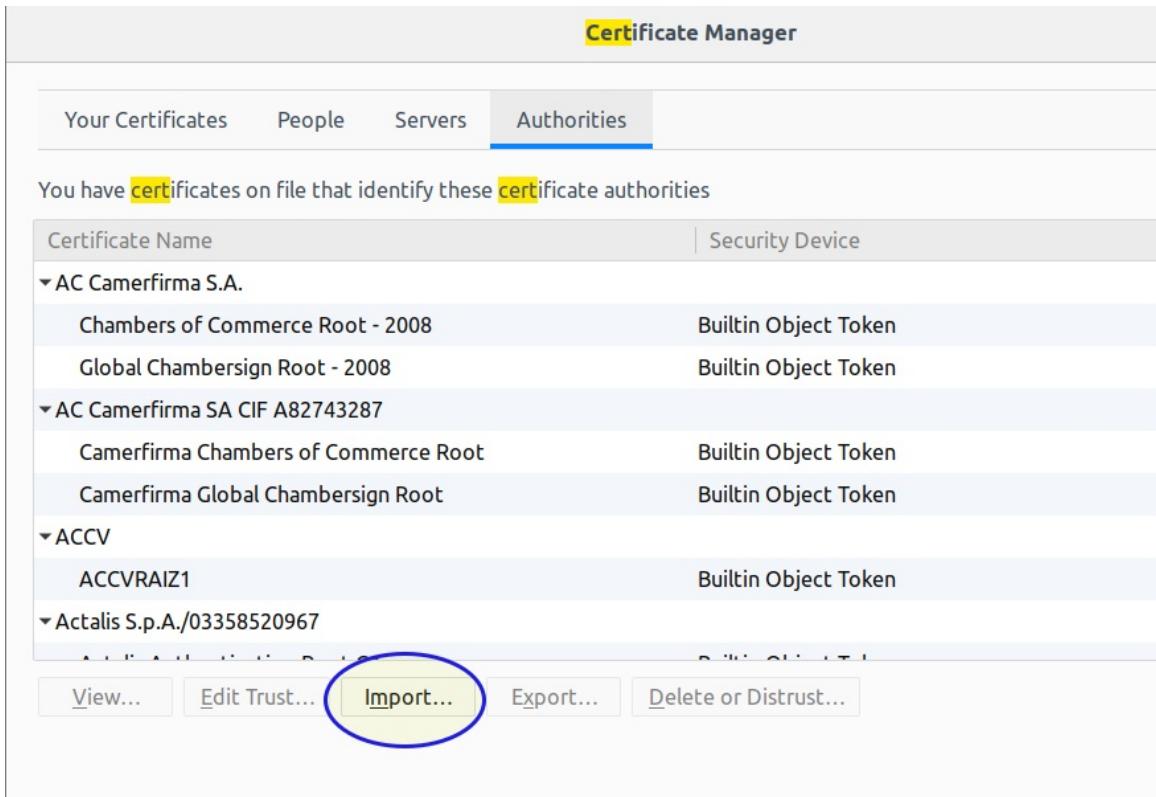
4. Click on "CA Certificate" link to download the "cacert.der" file.



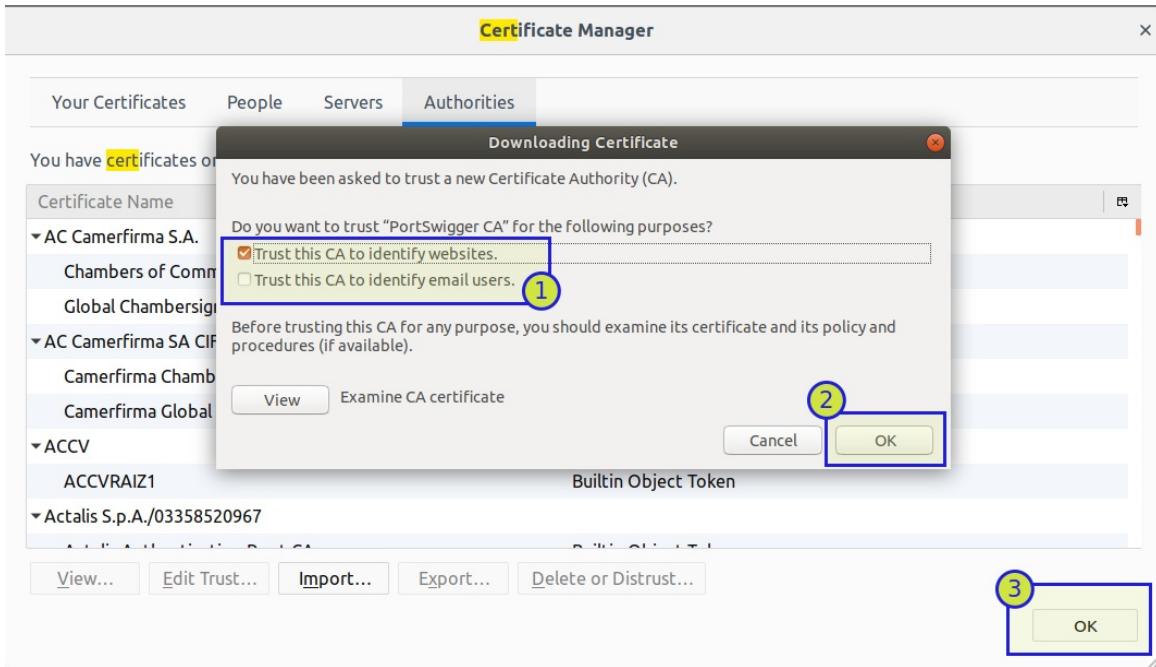
5. In the Firefox browser, go to "Preferences", search for the term "certificate", and click on "View Certificates" button.



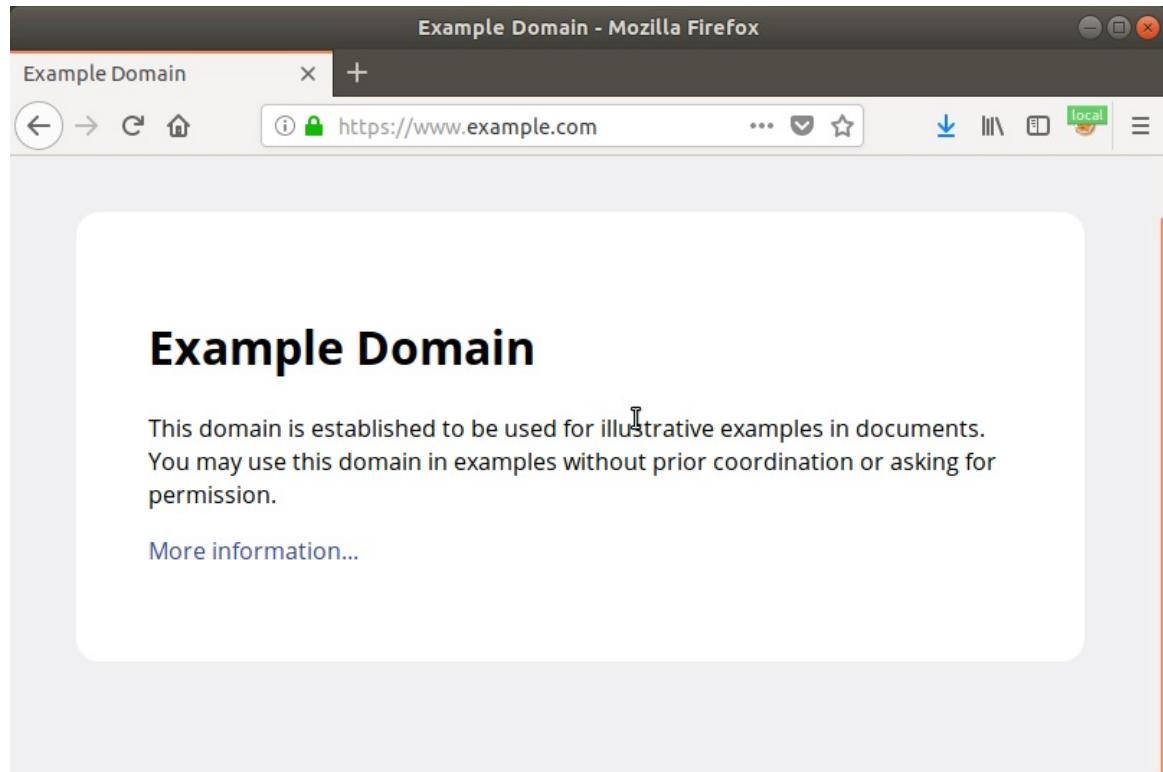
6. In the "Certificate Manager" window, click on "Import" button and select the downloaded "cacert.der" file.



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

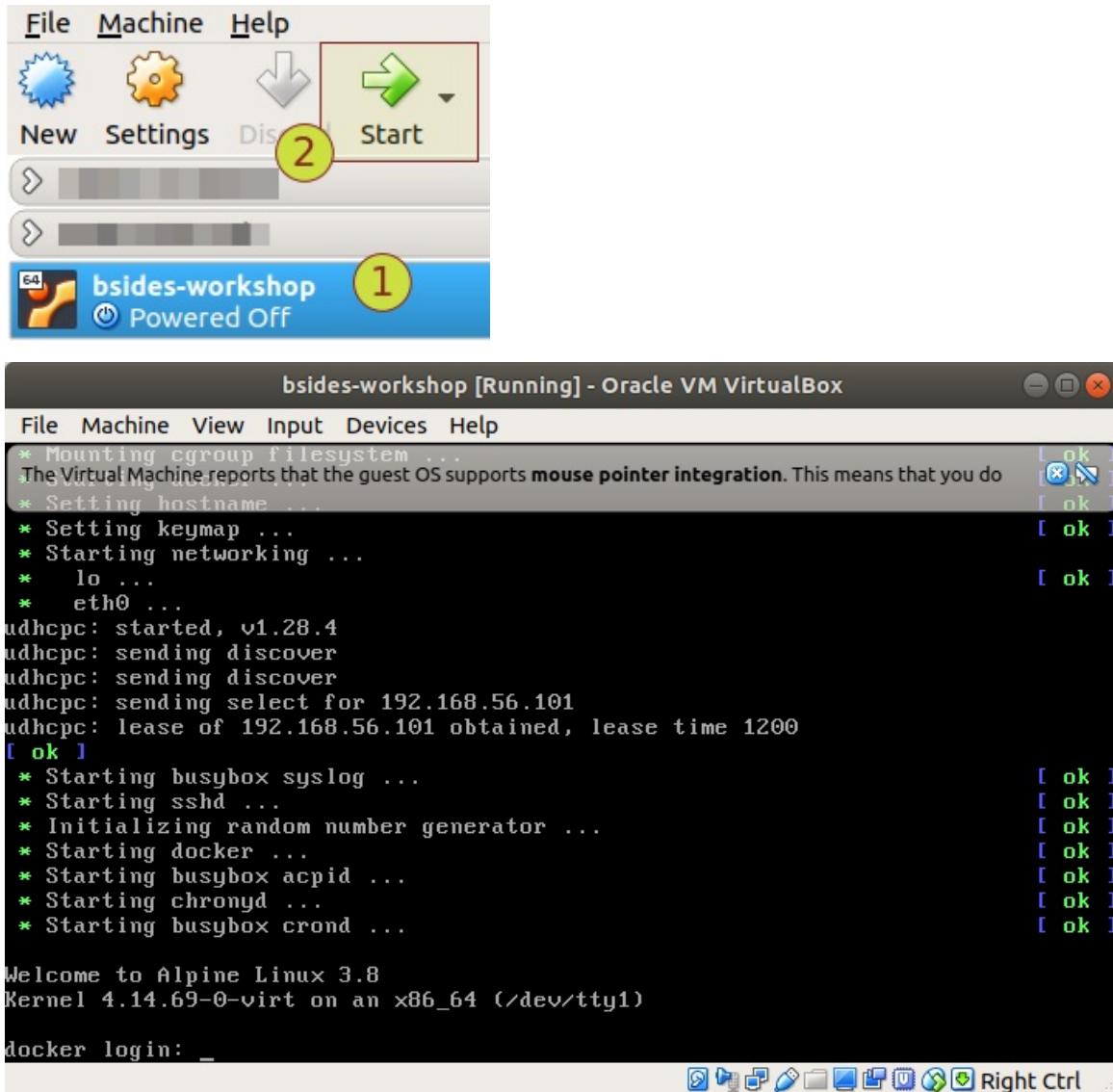


8. Access secure websites, e.g., "<https://www.example.com>", without encountering the "SEC_ERROR_UNKNOWN_ISSUER" error.



Access Security Shepherd Web Application

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



- Login to the Virtual Machine 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.

```
docker:~# ifconfig eth0
eth0      Link encap:Ethernet  HWaddr 08:00:27:71:47:57
          inet  addr: 192.168.56.101  Bcast:0.0.0.0  Mask:255.255.255.0
                  inet6 addr: fe80::a00:27ff:fe71:4757/64  Scope:Link
                      UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
                      RX packets:4 errors:0 dropped:0 overruns:0 frame:0
                      TX packets:17 errors:0 dropped:0 overruns:0 carrier:0
                      collisions:0 txqueuelen:1000
                      RX bytes:1830 (1.7 KiB)  TX bytes:2092 (2.0 KiB)

docker:~# _
```

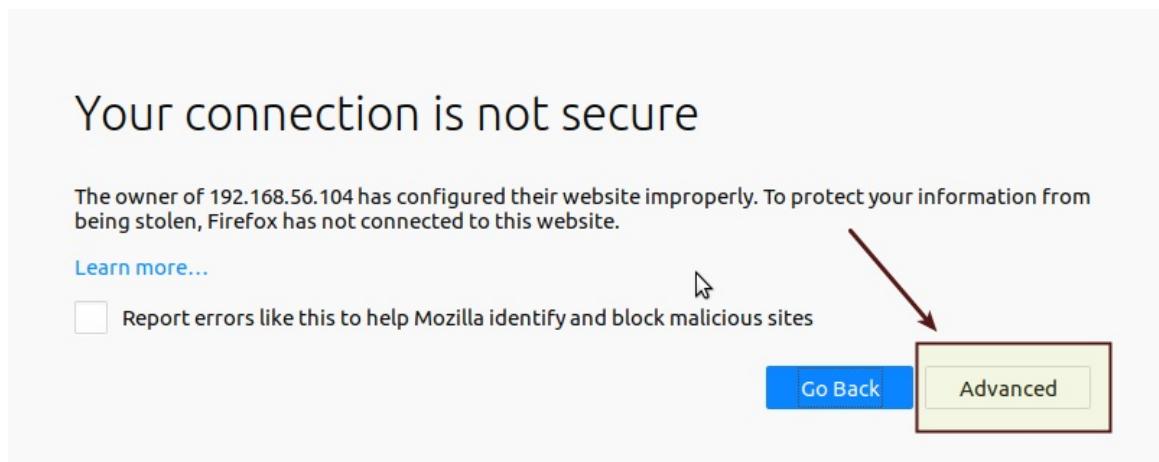
- In my case, I got the IP address as 192.168.56.101 .

5. Run following commands to start Security Shepherd application:

```
# cd SecurityShepherd/  
# docker-compose up -d
```

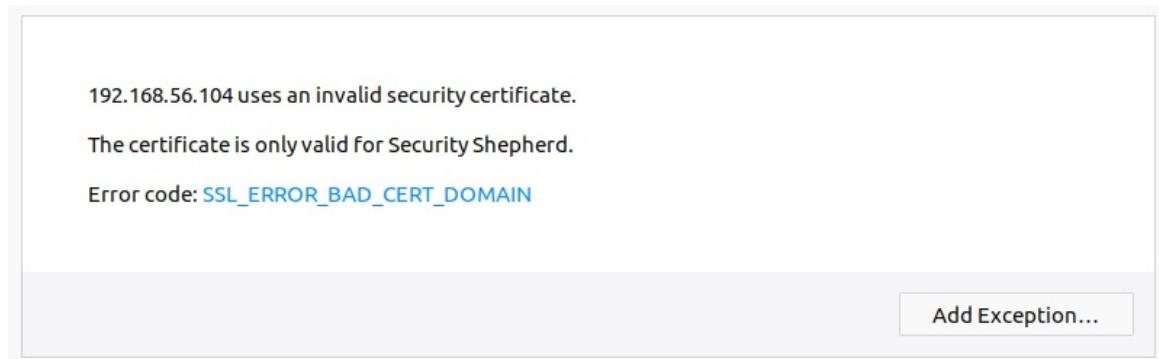
6. In Firefox, access the URL `http://192.168.56.101`.

7. If the browser throws an error saying "Your connection is not secure", click on the "Advanced" button to see error details.

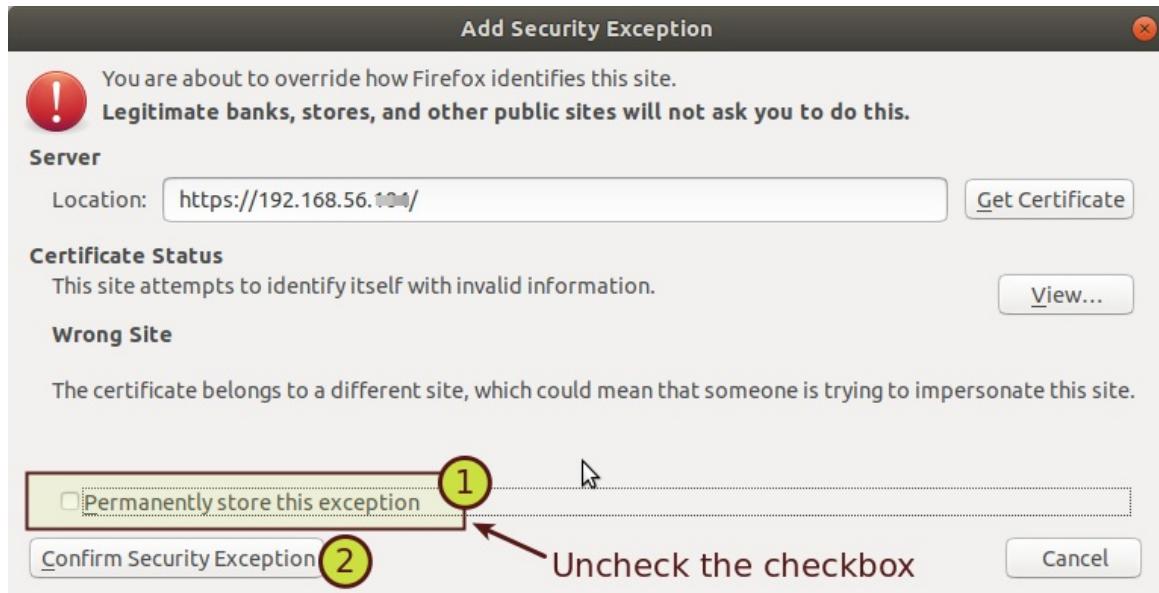


8. This time the error is not "SEC_ERROR_UNKNOWN_ISSUER", but it is "SSL_ERROR_BAD_CERT_DOMAIN" instead.

9. Click on "Add Exception" button.



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



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

The screenshot shows the 'Security Shepherd' login page. At the top, there is a language selector set to 'English'. The main title 'Security Shepherd' is displayed above a silhouette of a shepherd and his flock. Below the title, the word 'Login' is centered. A sub-instruction says 'Use your [Security Shepherd Credentials](#) to Login.' Another instruction says 'Register a [Security Shepherd Account](#) here!' Below these are two input fields: 'Username:' and 'Password:', each with a corresponding text input box. A 'Submit' button is located below the password field. At the bottom of the page, there are two links: 'Do you need a Proxy?' and 'About Security Shepherd'.

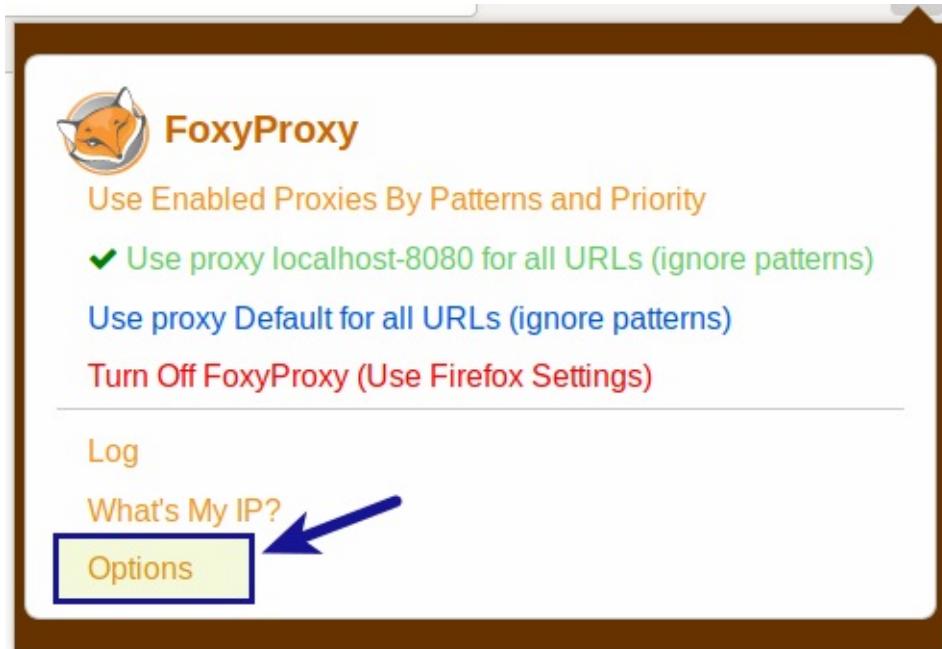
Getting Rid of Unnecessary Browser Traffic

1. In Burp, go to "Target" > "Site map" tab.
2. 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. A table displays network traffic entries:

Host	Method	URL	Params	Status	Length
http://detectportal.firefox.com	GET	/success.txt		200	379
https://profile.accounts.mozilla.com					
https://shavar.services.mozilla.com					
https://sync-641-us-west-2.sync.services.mozilla.com					
https://token.services.mozilla.com					
https://webextensions.settings.services.mozilla.com					

3. 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
4. In your Firefox browser, click on "FoxyProxy" icon and select "**Options**"



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



6. 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](#)

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

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

- 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

Use Enabled Proxies By Patterns and Priority

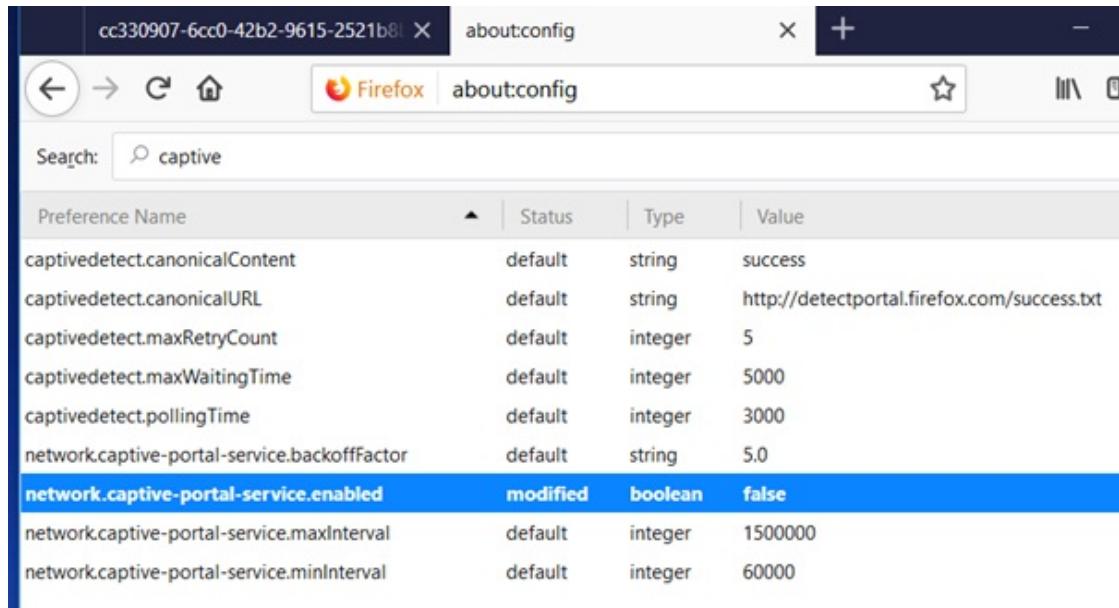
Synchronize settings: On ?

localhost-8080 127.0.0.1

Default

- Access `about:config` in the address bar of your Firefox browser.
- Disable following browser configuration by double-clicking on the respective preference name.

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



A screenshot of the Firefox browser's "about:config" page. The search bar at the top contains the text "captive". A table below lists various preferences related to captive portals. One preference, "network.captive-portal-service.enabled", is highlighted with a blue background, indicating it has been modified. The table has columns for Preference Name, Status, Type, and Value.

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](#)

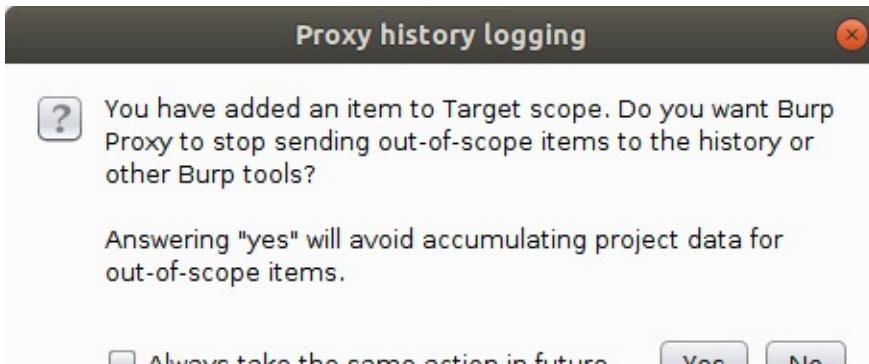
Filter: Hiding not found items; hiding CSS, image and general binary content; hiding 4xx responses; hiding empty folders

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

```

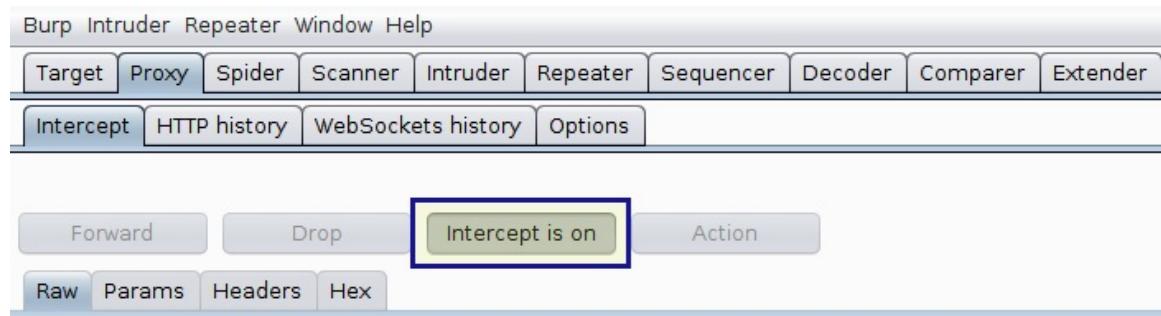
3. Switch to "Proxy" > "Websockets history" tab.

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 URL is https://192.168.56.101/login.js. The page has a dark background with a silhouette of a shepherd and a sheep. The title 'Security Shepherd' is at the top, followed by a large 'Login' button. Below it, text says 'Use your Security Shepherd Credentials to Login.' and 'Register a [Security Shepherd Account](#) here!'. A form is present with fields for 'Username' (containing 'b33f123') and 'Password' (containing '.....'). A 'Submit' button is at the bottom right. The entire form area is highlighted with a blue box and numbered 1, 2, and 3. The number 1 is next to the username field, 2 next to the password field, and 3 next to the submit button.

- 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.

The screenshot shows the Burp Suite interface in the Proxy tab. A POST request is displayed:

```

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

```

The request body contains the parameters:

login= [REDACTED] 123 & **pwd=** [REDACTED] 12345 & **submit=Submit**

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

The screenshot shows the Burp Suite interface in the Proxy tab after tampering. The request body now contains:

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 Burp Suite interface with the "HTTP history" sub-tab selected. The table displays a list of intercepted items:

#	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

Below the table, there are tabs for "Request" and "Response". At the bottom, there are buttons for "Raw", "Params", "Headers", and "Hex".

- 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

4. 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 context menu is open over a selected request (POST /login). The menu options include:

- Add comment (highlighted)
- Ctrl+I
- Delete item
- Clear history
- Copy URL
- Send to Repeater
- Ctrl+R
- Copy as curl command
- Copy links
- Save item
- Proxy history help

The selected request is highlighted in yellow. The status bar at the bottom indicates '0 matches'.

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 context menu is open over a selected request (POST /login). A 'Comment' dialog box is open in the foreground, prompting the user to 'Enter a comment' with the placeholder 'Enter something meaningful here...'. The selected request is highlighted in yellow. The status bar at the bottom indicates '0 matches'.

8. If you wish to forward an interesting request to Scanner, Repeater, or 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

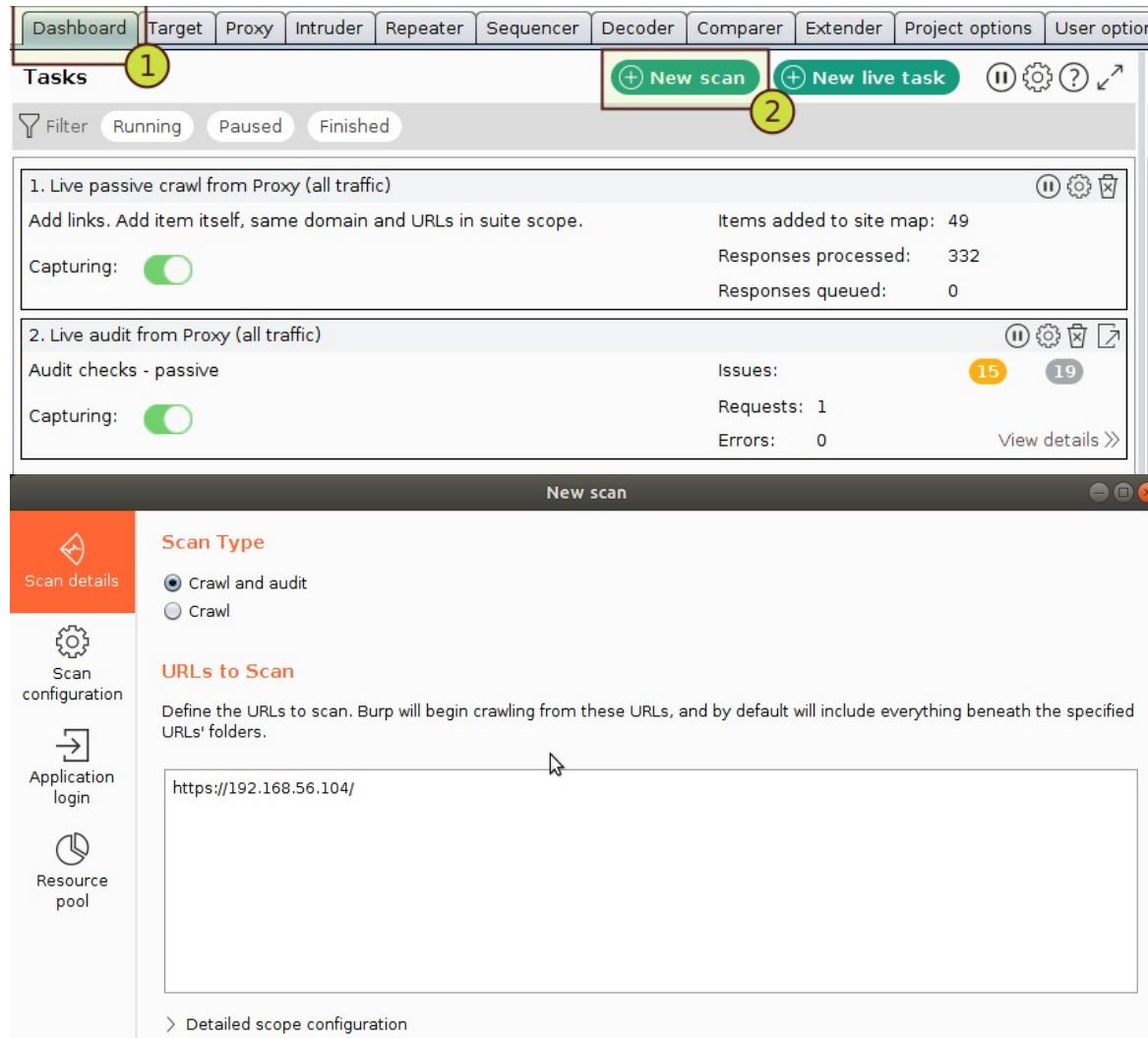
0100101 Firefox/61.0

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

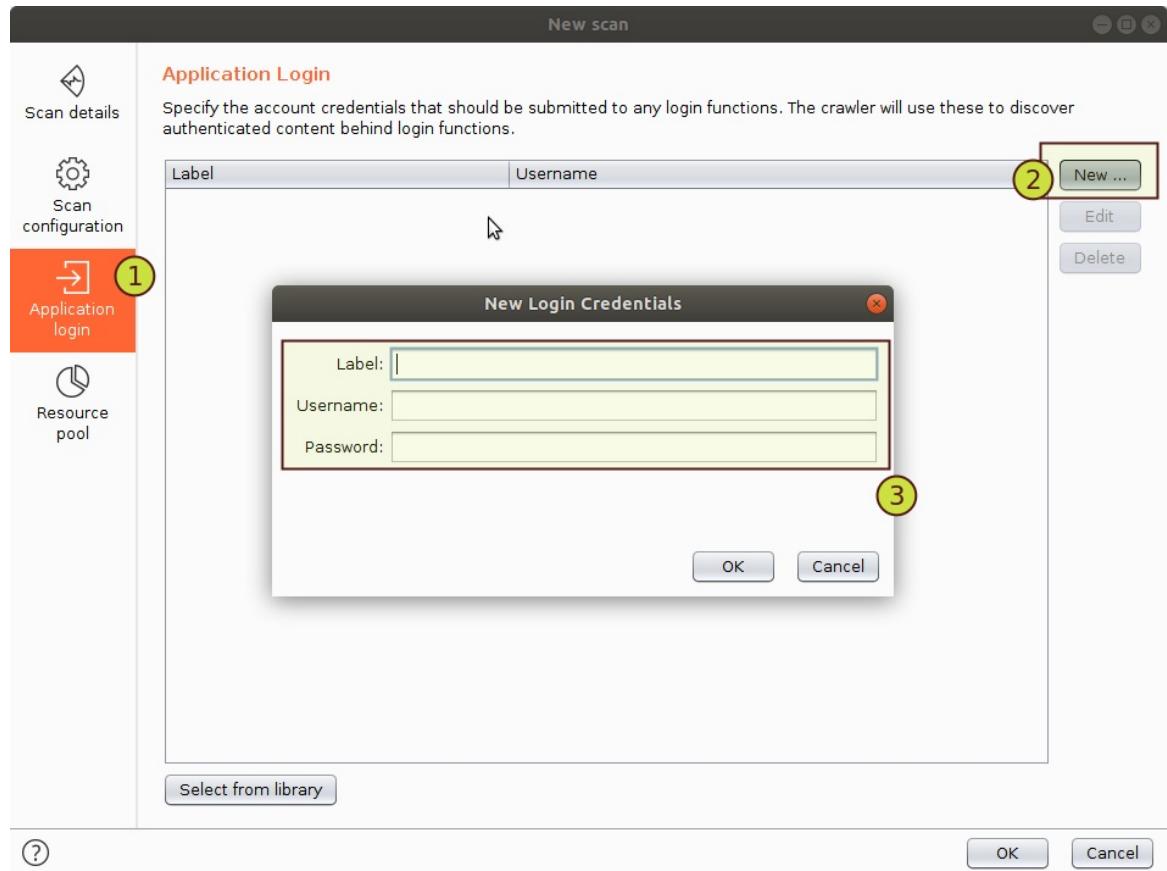
Scanner (10 Minutes)

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](#)

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.

Request Headers

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

- Update Content-Length header
- Set Connection: close

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):	<input checked="" type="radio"/> Fixed: 0 <input type="radio"/> Variable: start 0 step 30000
Start time:	<input checked="" type="radio"/> Immediately <input type="radio"/> In 10 minutes <input type="radio"/> 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.

The screenshot shows the Burp Suite interface with the "Intruder" tab selected. A single request is listed under the "Intercept" tab. The request is a POST to the login page. The "Action" dropdown is set to "Action". The "Intercept is on" checkbox is checked. The "Raw" tab shows the request body: "login=someuser&pwd=somepassword&submit=Submit". The "Params" tab shows the parameters: "username=someuser" and "password=somepassword". The "Headers" tab shows the headers: "Content-Type: application/x-www-form-urlencoded", "Content-Length: 45", "Accept-Encoding: gzip, deflate", "Referer: https://192.168.56.104/login.jsp", "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", and "Connection: close". The "Hex" tab shows the raw hex and ASCII representation of the request. To the right of the Burp interface, the actual "Security Shepherd" web page is displayed, showing the "Login" form with the same fields and values.

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.

Start attack

Payload set: 1 Payload count: 0

Payload type: Simple list Request count: 0

Payload Options [Simple list]

Add from list ...

Usernames
Passwords
Short words
a-z
A-Z

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

21. 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

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

Intruder attack 4

Attack Save Columns

Results Target Positions Payloads Options

Filter: Showing all items

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

23. 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:

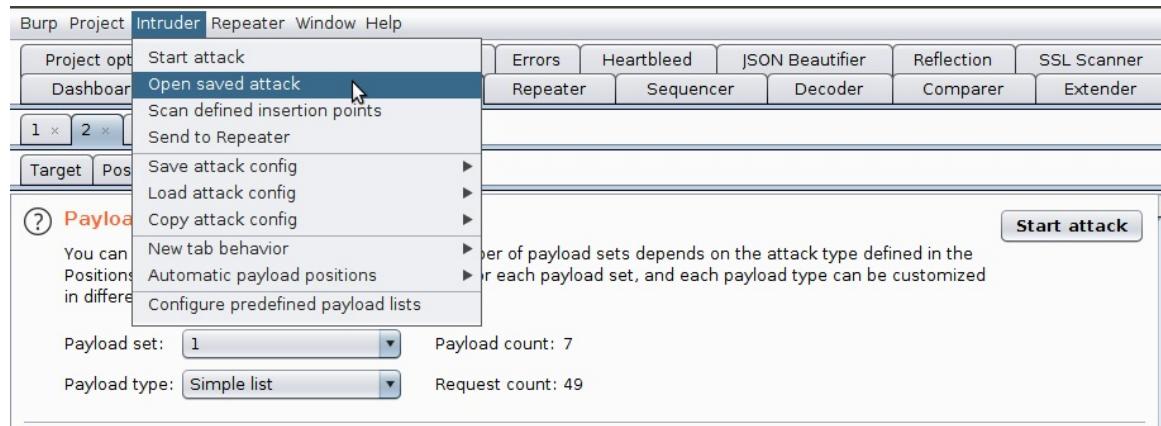
New Password:

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.

Scoreboard

Completed

Insecure Direct Object References

Submit Result Key Here...

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.

Get Next Challenge

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

User: Guest

Age: 22
Address: 54 Kevin Street, Dublin
Email: guestAccount@securityShepherd.com
Private Message: No Private Message Set

HTTP history | WebSockets history | Options

Request to https://192.168.56.104:443

Forward Drop Intercept is on Action Comment this item

Raw Params Headers Hex

```
POST /lesson/insecureDirectObjectReferences 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/lesson/insecureDirectObjectReferences
Content-Type: application/x-www-form-urlencoded
X-Requested-With: XMLHttpRequest
Content-Length: 14
Cookie: JSESSIONID=21BFF9296030364CC68BDD83667F06B;
token=-60022310048214894798442406549964184552; JSESSIONID3="POwG1rhMoYhh6ExoSXSjDw=="
Connection: close
username=guest
```

Security Shepherd

Submit Result Key Here...

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.

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

Loading...

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><br><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' 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' alt='Copy to clipboard'></button></span><p>&nbsp;</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

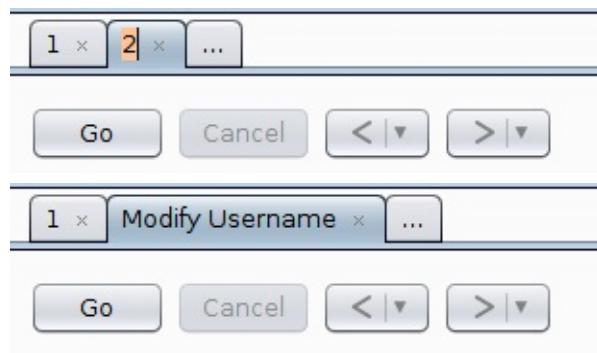
18. To see the previously triggered requests, click on the back arrow symbol.



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 orange. 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:

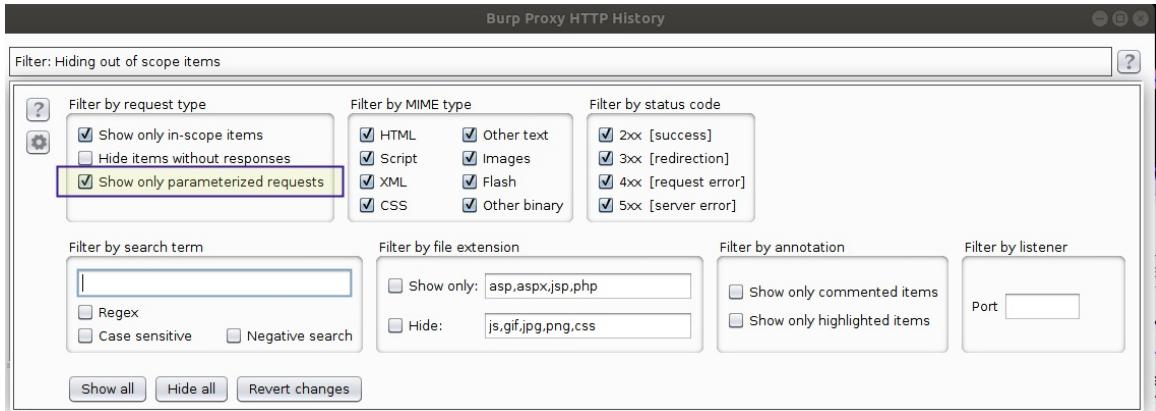
- Remove from scope
- Spider from here
- Do an active scan
- Do a passive scan
- Send to Intruder (Ctrl+I)
- Send to Repeater (Ctrl+R)
- Send to Sequencer
- Send to Comparer (request) (Ctrl+Alt+1)
- Send to Comparer (response) (Ctrl+Alt+2)
- Show response in browser
- Request in browser (▶)
- Engagement 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 'Params' field contains the value `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=91625F18C1CC68...	
POST	/login	✓		302	345	text			✓	192.168.56.104	JSESSIONID=9020EEA0F1FC93...	
POST	/login	✓		302	344	text			✓	192.168.56.104	JSESSIONID=21BF929603036...	
POST	/lessons/fdb94122d0f032821019c7edf09dc62ea21e...	✓	✓	200	266	text			✓	192.168.56.104	JSESSIONID=0CAF93093B3944...	
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		

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.

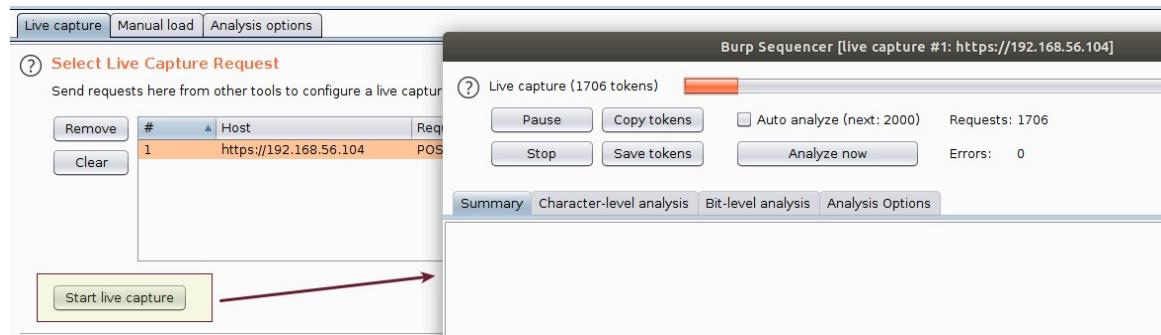
Cookie:

Form field:

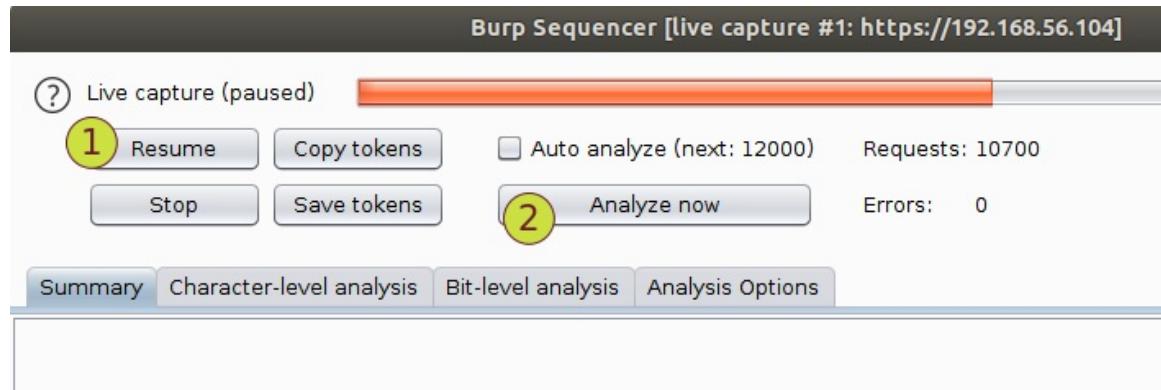
Custom location:

Configure

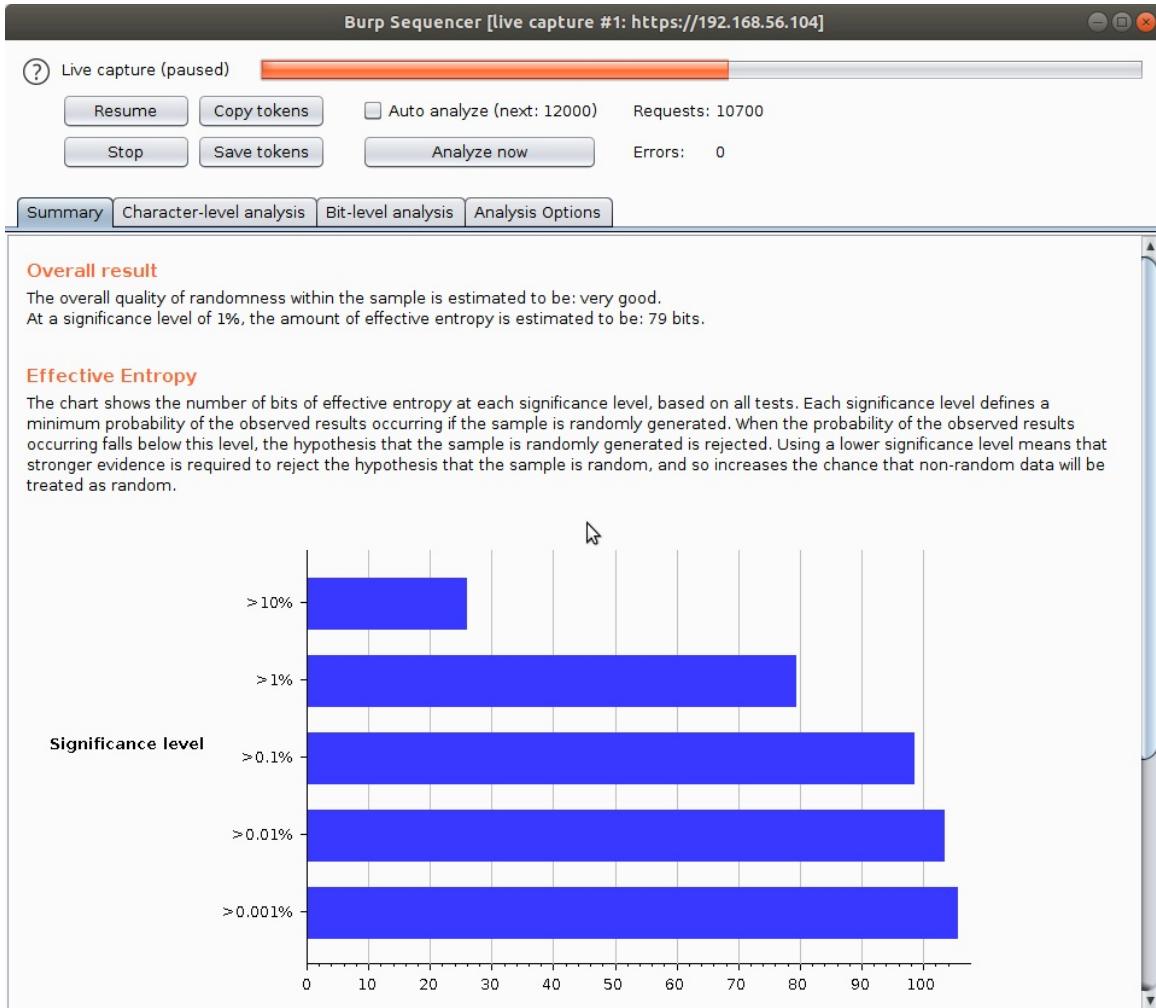
12. Click on the "Start live capture" button.



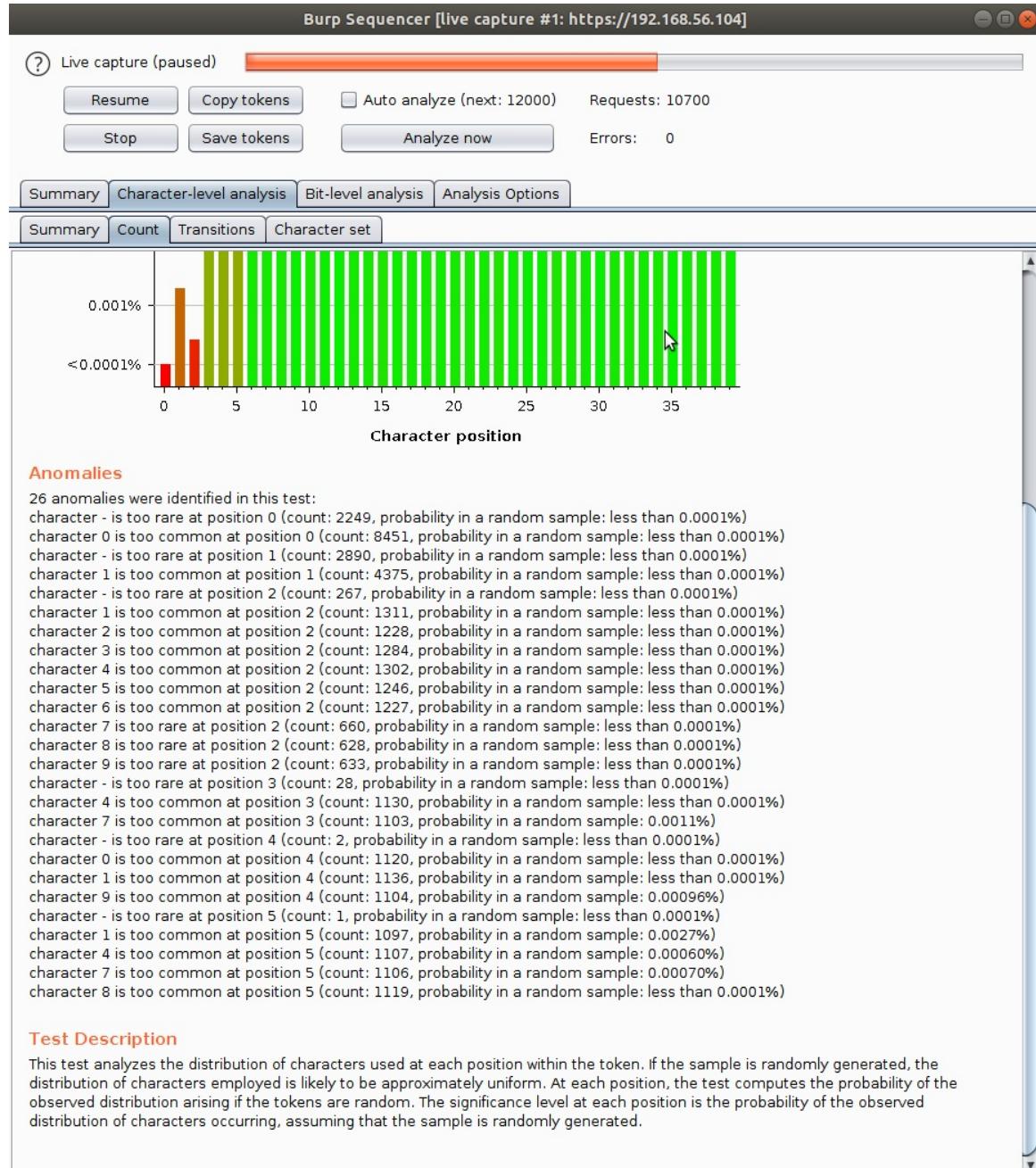
13. When a few hundred tokens have been obtained, pause the live capture session and click the "Analyze now" button.



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.



16. 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.

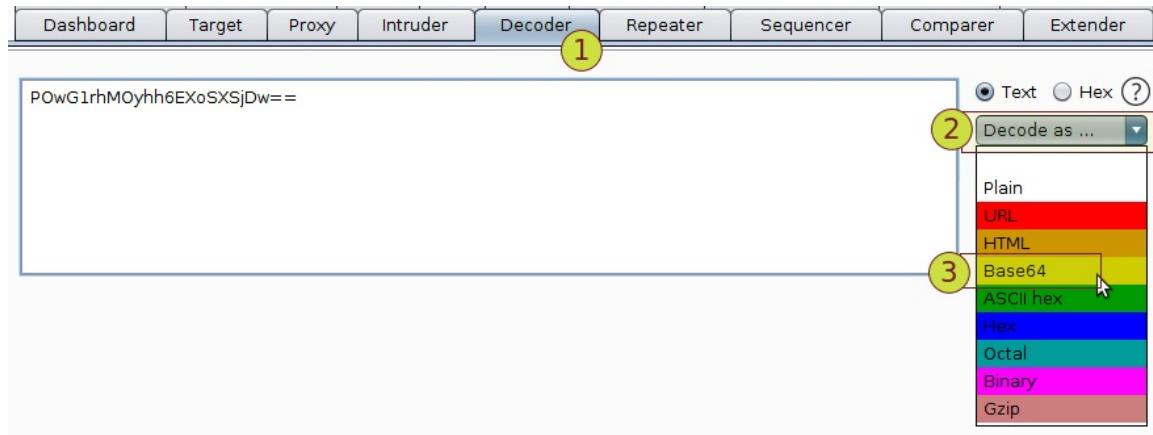
The screenshot shows the Burp Suite interface with the 'HTTP history' tab selected. The 'Filter' dialog is open, showing various filtering options. The 'Filter by search term' section has a green highlight around the search field containing the text '=='.

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 POST request to `/login` is selected. The request body contains a base64 encoded string. A context menu is open over the row, with the 'Send to Decoder' option highlighted.

#	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	/js/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 interface remains largely the same, with the 'Text' radio button selected in both the original and decoded sections, and the 'Decode as ...' dropdown set to 'Text'.

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

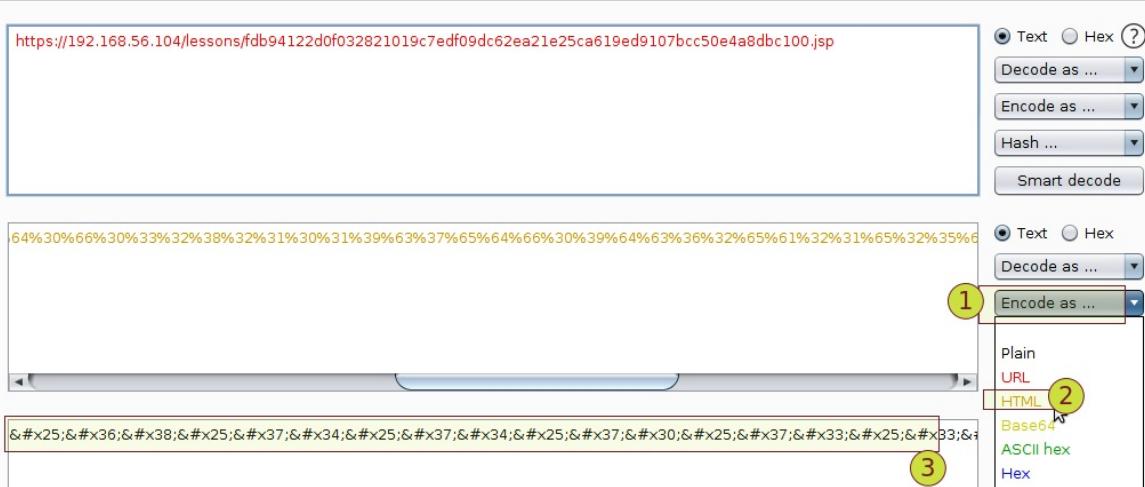
```
https://192.168.56.104/lessons/fdb94122d0f032821019c7edf09dc62ea21e25ca619ed9107bcc50e4a8dbc100.jsp
```

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

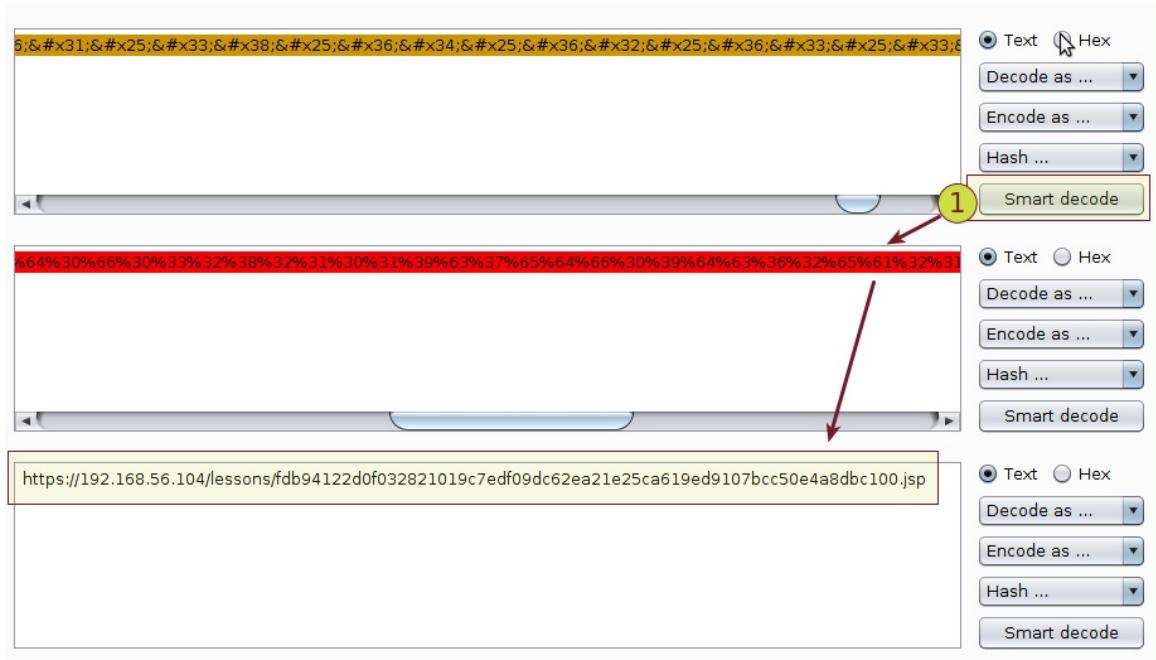
This screenshot shows the 'Encode as ...' dropdown menu from the previous step. The 'URL' option is highlighted with a yellow box, indicating it has been selected.

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.

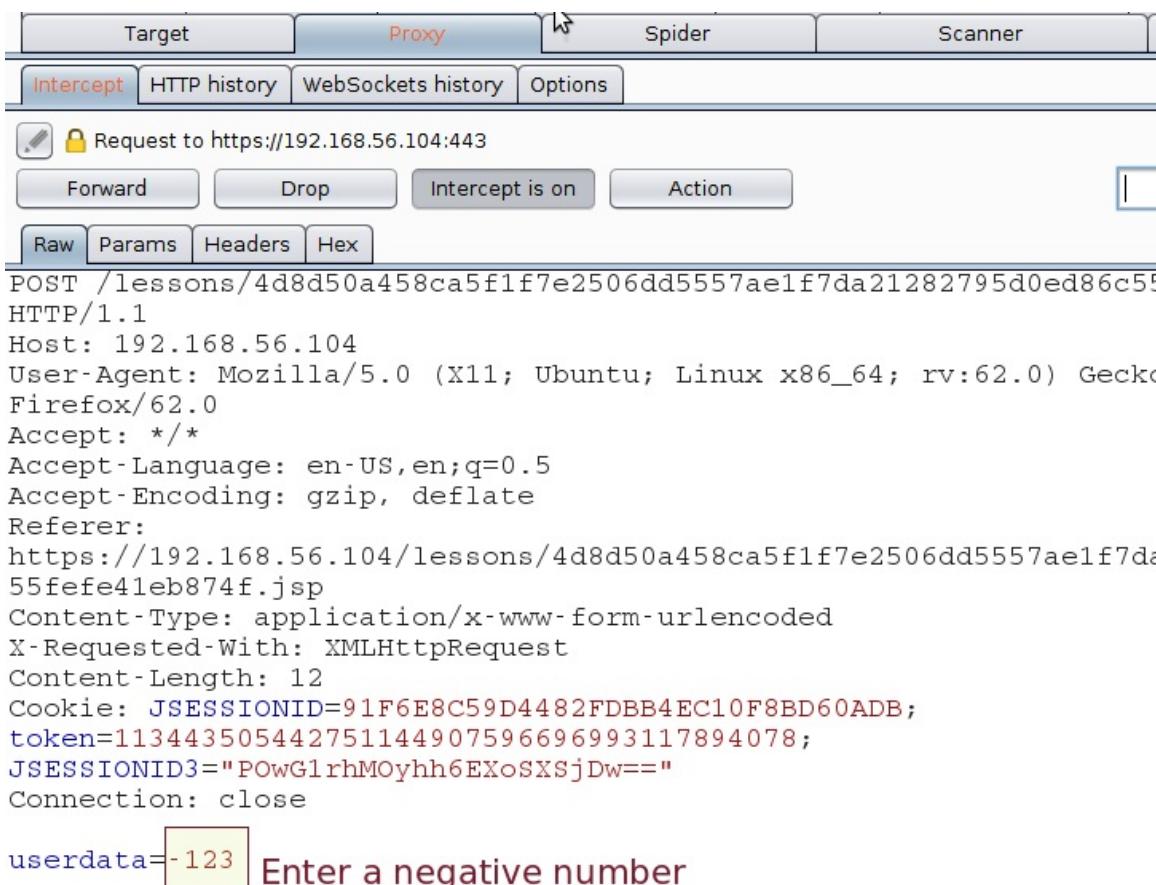
Word compare of #1 and #2 (11 differences)

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.

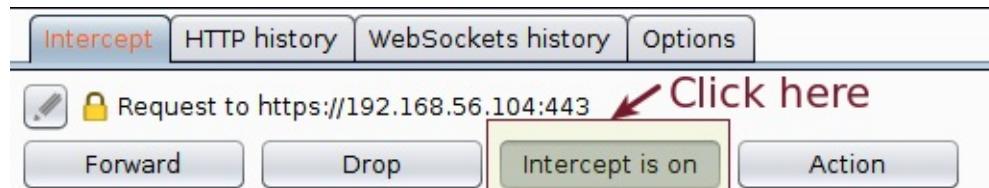
```

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:<a><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%'>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 heading says 'BApp Store' and a sub-heading states '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 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 prominently displayed 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 for 'Heartbleed', 'JSON Beautifier', 'Reflection', 'SSL Scanner', and 'Logger' (which is the active tab and highlighted in blue). Below the navigation bar is a table with 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' (selected), 'Response', 'Raw', 'Hex', and a search bar with the placeholder 'Type a search term' and a result count of '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>									

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 to 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, and also 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 that we deliver to our clients always follow the 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 address the specific problem a client seeks to solve and must include actionable insights to enable them to achieve this
3. Never be purely automated; whenever we are testing for security issues, our reports and output always include significant, expert, human inputs to give the greatest possible value for our clients

In addition to client-facing work, our technical teams 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>

