

# Android Mobile Pentest 101

*© tsug0d, September 2018*

# Lecture 5 – Basic Dynamic Analysis

Goal: Known how to use Burp Suite

# Introduction

- **Dynamic analysis** is the testing and evaluation of a program by executing data in real-time.
- The objective of Dynamic Analysis is to find errors in a program while it is running, rather than by repeatedly examining the code offline.
- We are going to use Burp Suite to do our job

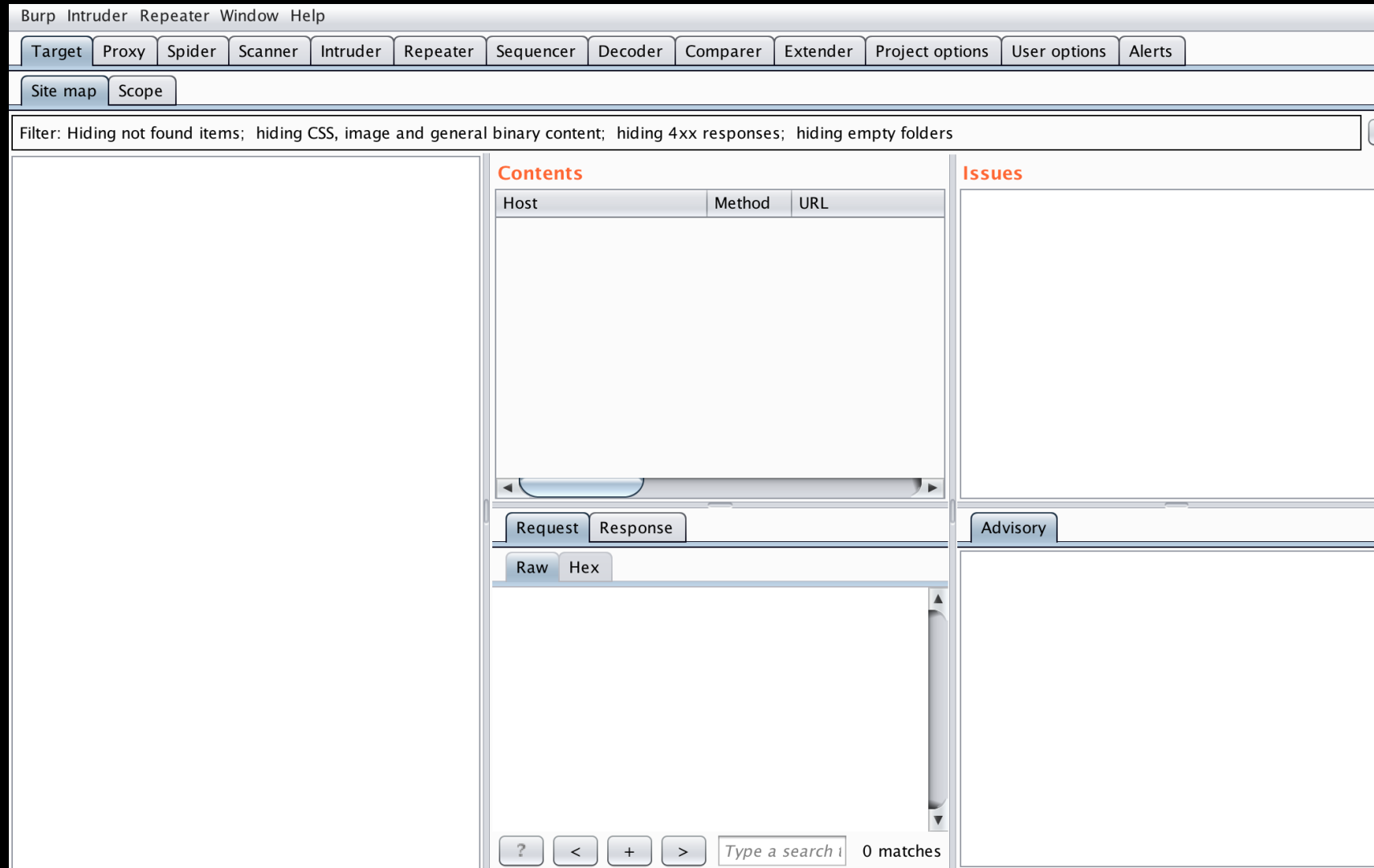


# Installation

- **Burp Suite** is a Java based Web Penetration Testing framework. It has become an industry standard **suite** of tools used by information security professionals
- To install Burp Suite, come to:  
<https://portswigger.net/burp/communitydownload>
- Download the file that suit for you, in this lecture, i'll use the .jar file

# How-to-use

- Open Burp Suite:



# How-to-use

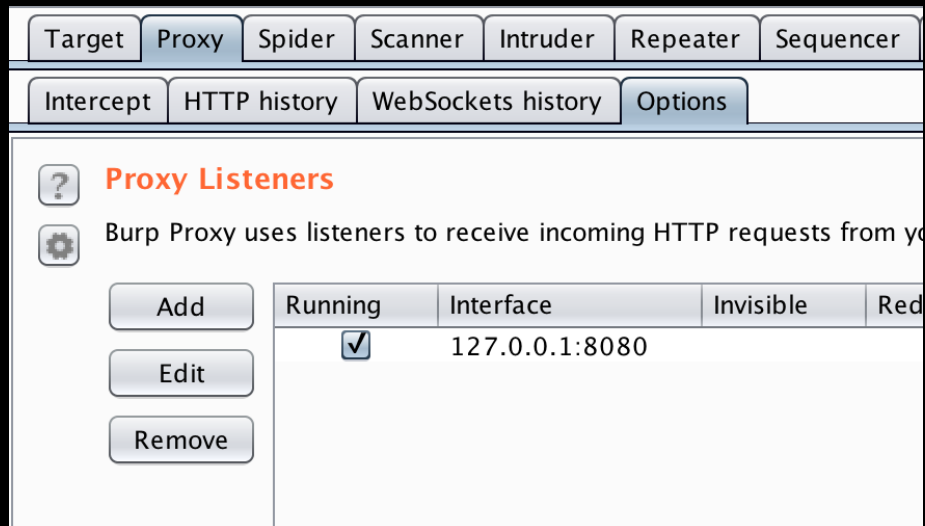
- I'm not going to introduce all tab of Burp, instead I'll focus on some important tab for mobile pentest.
- First we need to setup our Device to proxy via Burp, it means any request from device to the network will be intercept by Burp Suite
- We check the ip address of our device to determine the network range:

```
🍏 ~/Desktop/mobile/tools/ ./adb devices
```

```
List of devices attached
```

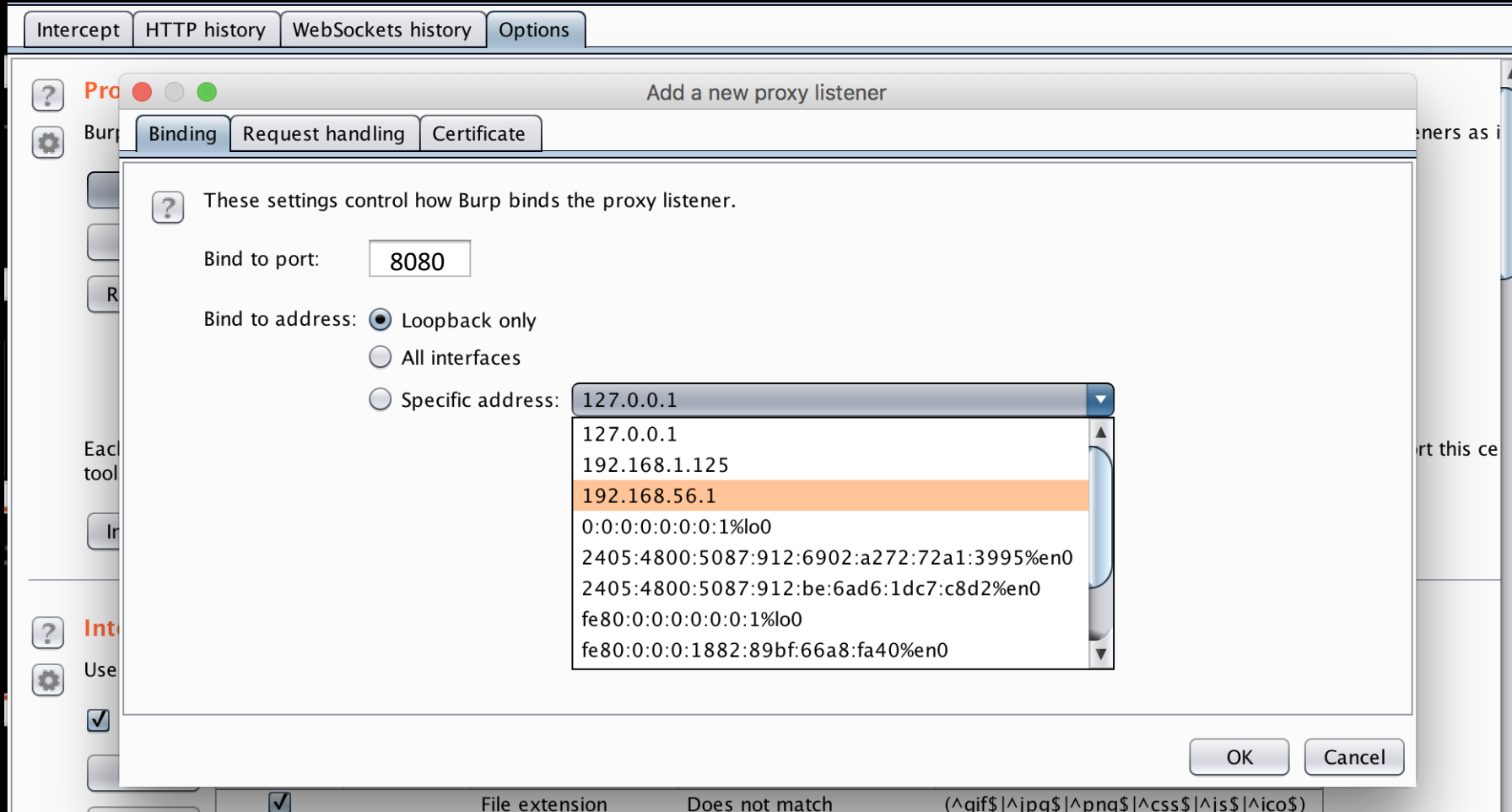
```
192.168.56.101:5555      device
```

- Then go to **Burp Suite -> Proxy -> Options**



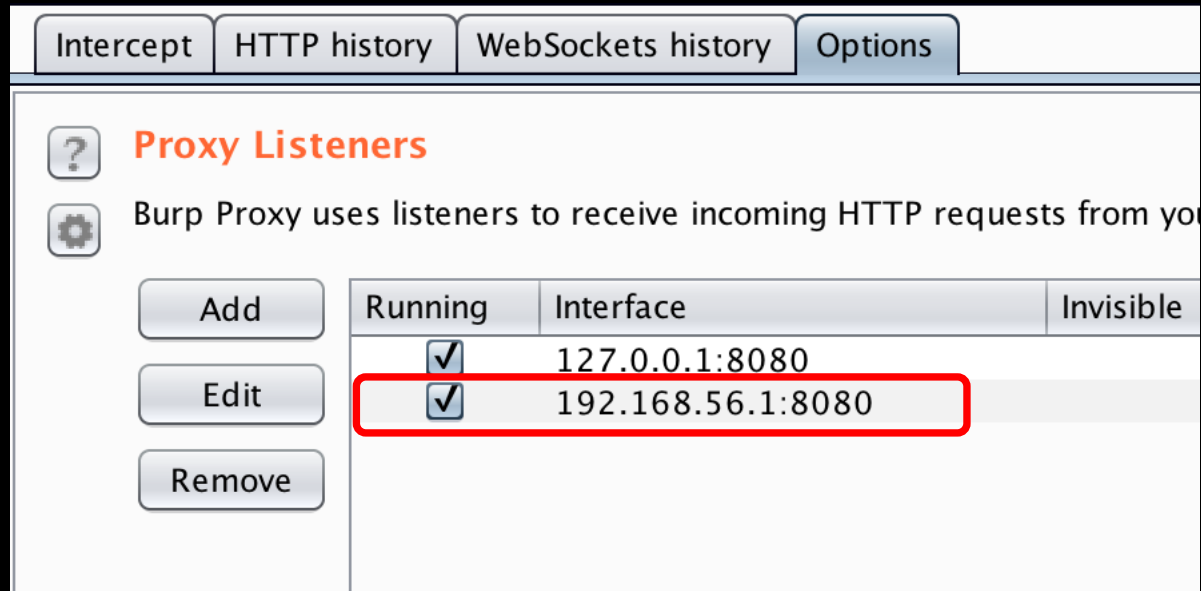
# How-to-use

- Click Add, then choose the ip address in the same network range of device to become a listener



# How-to-use

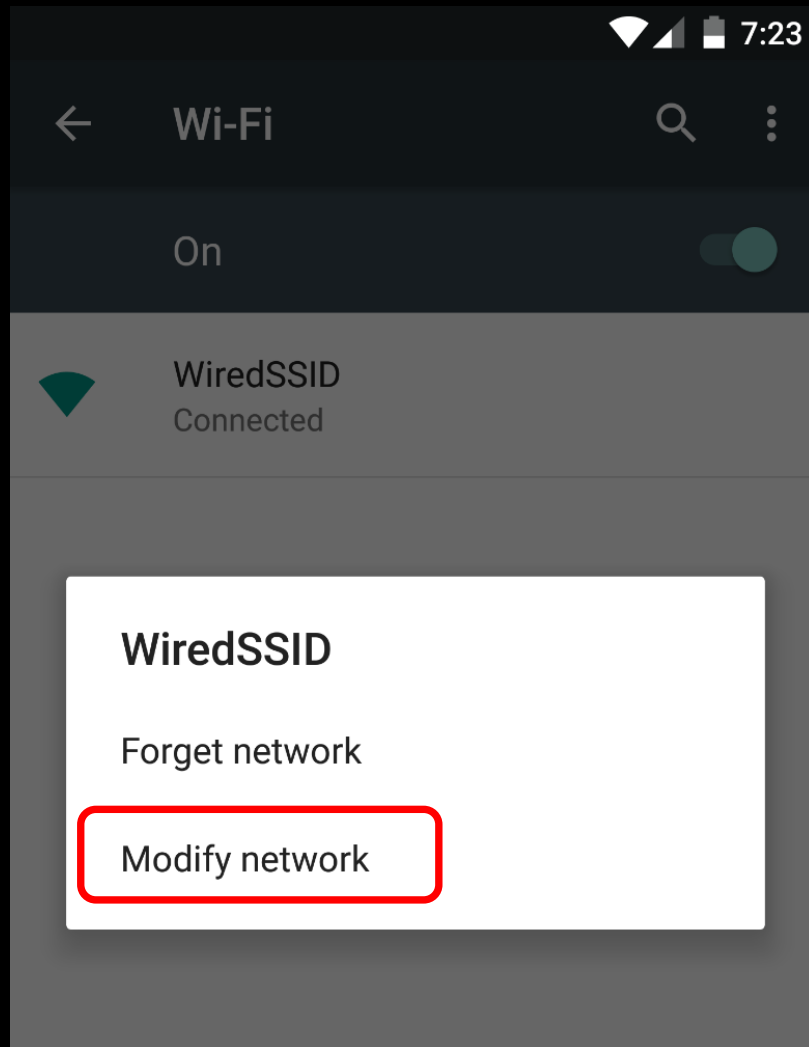
- In **Proxy Listeners**, our new Listen interface appears, tick on running





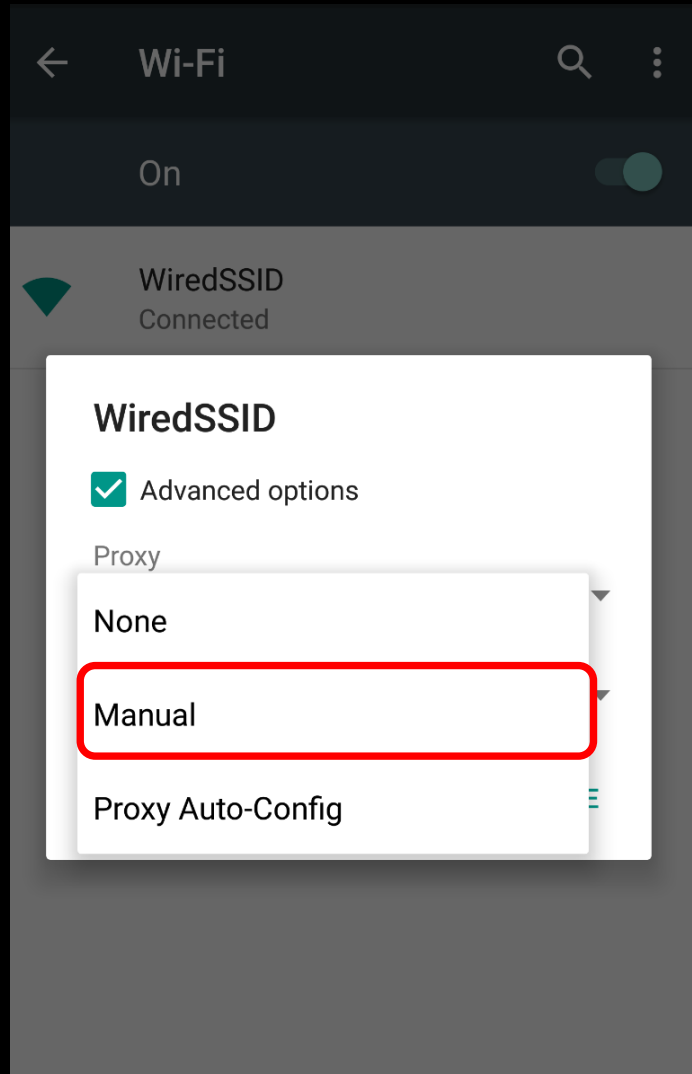
# How-to-use

- Now go to our **Device -> Settings -> Wifi**, click and hold the **wifi** we saw



# How-to-use

- Choose **Modify network**, tick on **Advanced options**, from **Proxy scroll**, choose **Manual**



# How-to-use

- Fill-in the listener that we've created above -> **Save**

**WiredSSID**

☒ Advanced options

Proxy  
Manual

The HTTP proxy is used by the browser but may not be used by the other apps.

Proxy hostname  
192.168.56.1

Proxy port  
8080

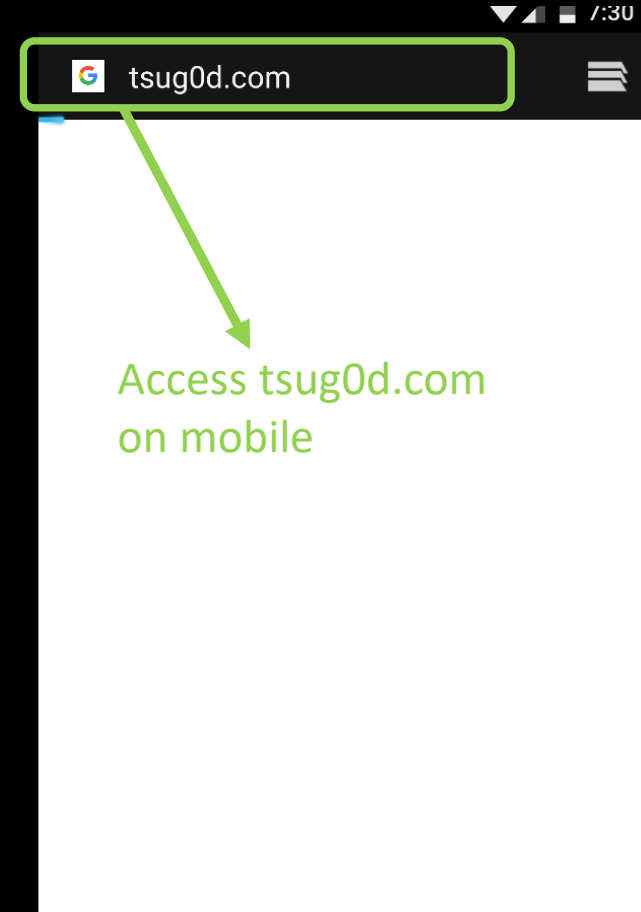
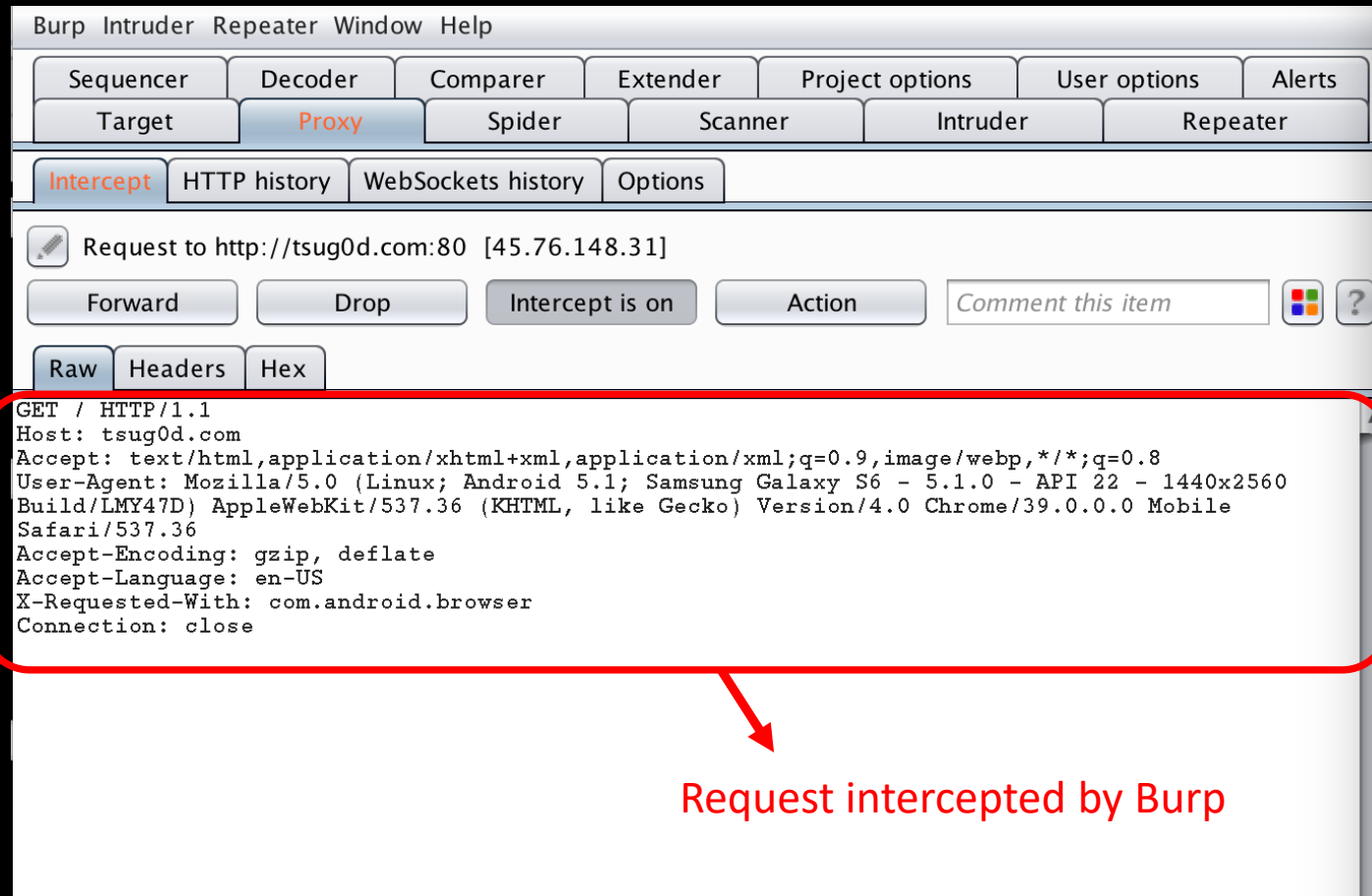
Bypass proxy for  
example.com,mycomp.test.com,l

IP settings  
DHCP

CANCEL SAVE

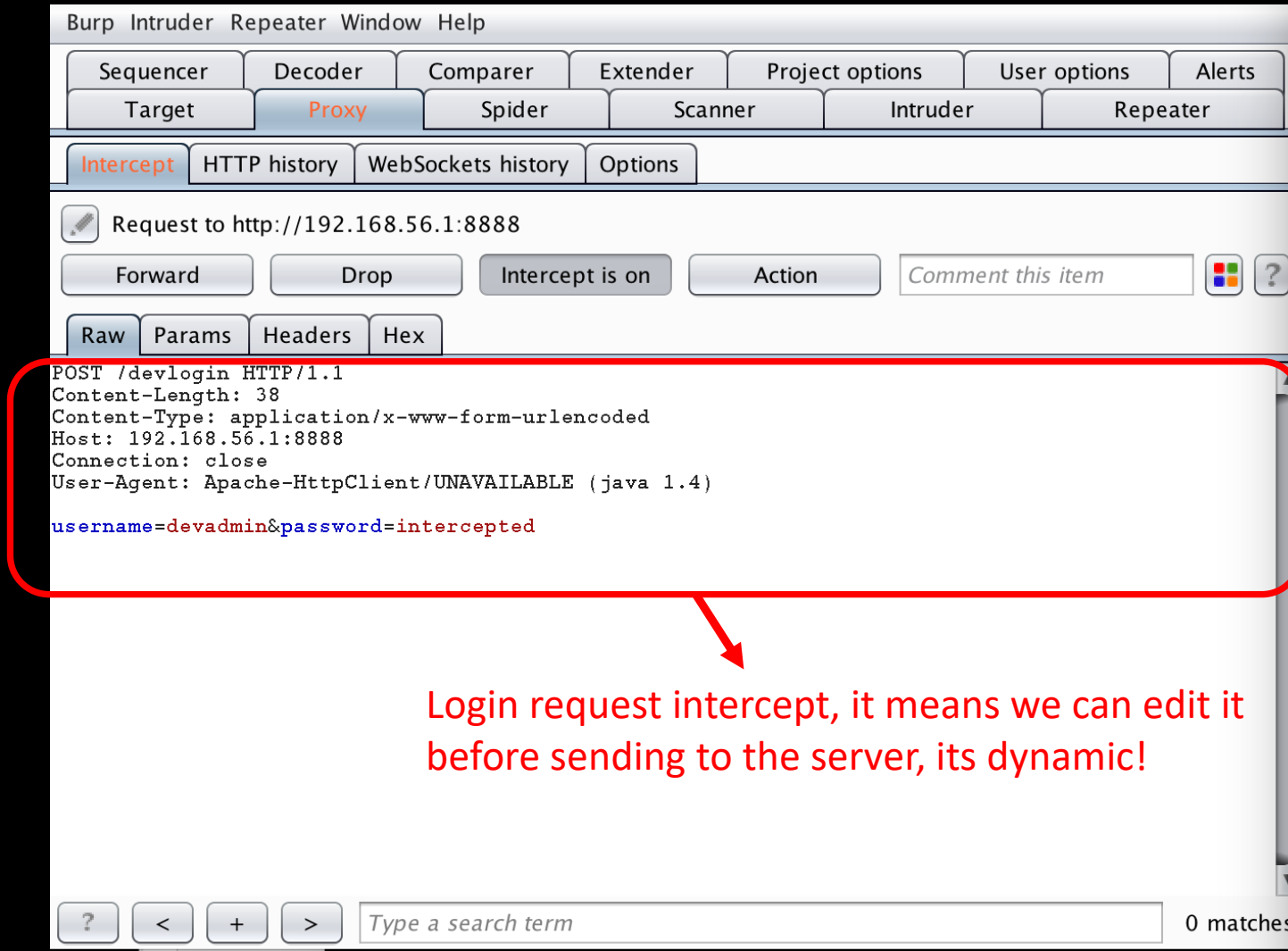
# How-to-use

- We are almost done the setup, try to surf web via mobile browser:



# How-to-use

- Now trying with our app:



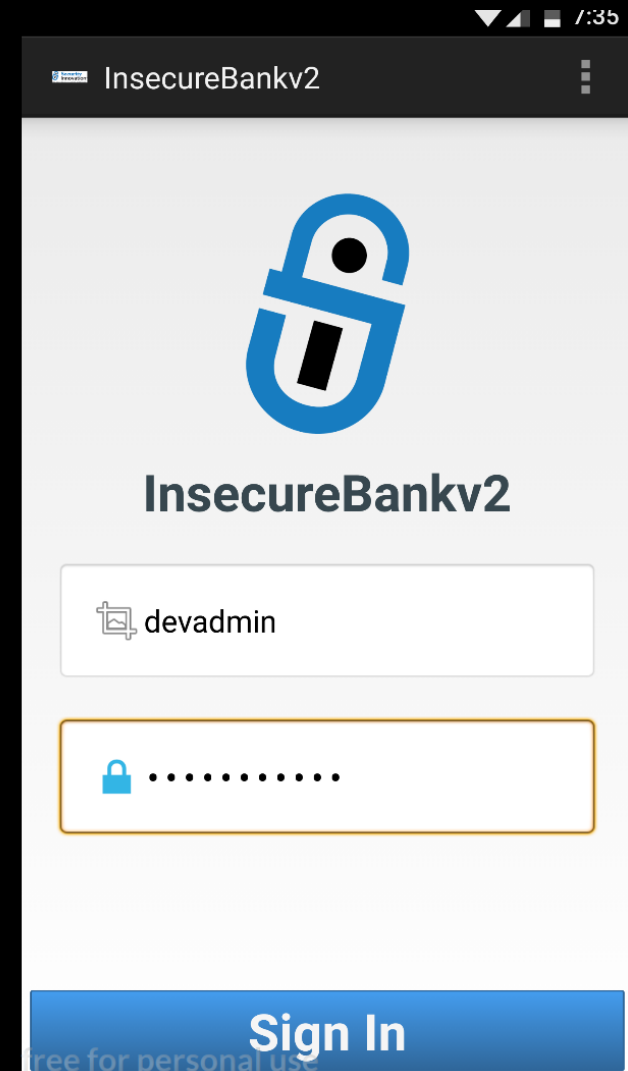
The screenshot shows the Burp Suite interface with the 'Intercept' tab selected. A red box highlights the intercepted request details, which are as follows:

```
POST /devlogin HTTP/1.1
Content-Length: 38
Content-Type: application/x-www-form-urlencoded
Host: 192.168.56.1:8888
Connection: close
User-Agent: Apache-HttpClient/UNAVAILABLE (java 1.4)

username=devadmin&password=intercepted
```

A red arrow points from the highlighted request body to the text below.

Login request intercept, it means we can edit it before sending to the server, its dynamic!



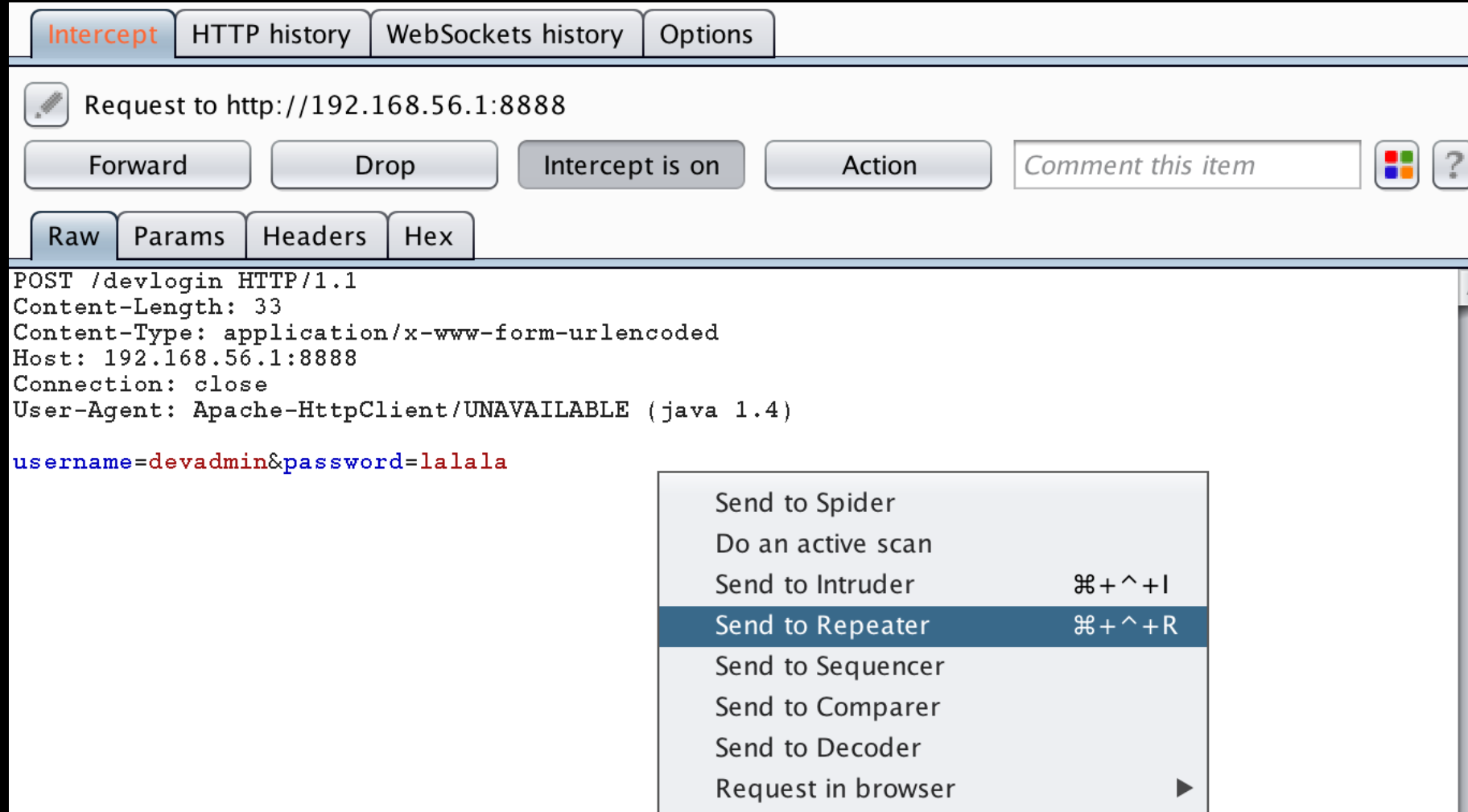
The screenshot shows the InsecureBankv2 mobile app login screen. It features a blue padlock icon, the text 'InsecureBankv2', and a login form with the following fields:

- Username field: devadmin
- Password field: masked with dots
- Sign In button

At the bottom, there is a link for 'Free for personal use'.

# More Burp

- Burp brings to you many feature, for example, if you wanna save time not request and intercept to edit each request, just intercept the first request and send it to **Repeater tab**



# More Burp

- Then testing on it 😊

The screenshot displays the Burp Suite interface with the 'Proxy' tab selected in the top toolbar. Below the toolbar, a tab bar shows '1 x ...'. A red rectangle highlights the 'Go' button in the toolbar. The main area is divided into two panels: 'Request' on the left and 'Response' on the right. The 'Request' panel has tabs for 'Raw', 'Params', 'Headers', and 'Hex', with 'Raw' selected. It shows an HTTP POST request to '/devlogin' with a 'Content-Type' of 'application/x-www-form-urlencoded' and a 'User-Agent' of 'Apache-HttpClient/UNAVAILABLE (java 1.4)'. The request body is 'username=devadmin&password=lalalanana'. The 'Response' panel has tabs for 'Raw', 'Headers', and 'Hex', with 'Raw' selected. It shows an HTTP 200 OK response with a 'Content-Type' of 'text/html; charset=utf-8' and a 'Server' of 'localhost'. The response body is a JSON object: '{"message": "Correct Credentials", "user": "devadmin"}'.

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

1 x ...

Go Cancel <|v >|v

**Request**

Raw Params Headers Hex

POST /devlogin HTTP/1.1  
Content-Length: 37  
Content-Type: application/x-www-form-urlencoded  
Host: 192.168.56.1:8888  
Connection: close  
User-Agent: Apache-HttpClient/UNAVAILABLE (java 1.4)  
username=devadmin&password=lalalanana

**Response**

Raw Headers Hex

HTTP/1.1 200 OK  
Content-Type: text/html; charset=utf-8  
Content-Length: 54  
Connection: close  
Date: Tue, 11 Sep 2018 18:40:34 GMT  
Server: localhost  
  
{"message": "Correct Credentials", "user": "devadmin"}

# More Burp

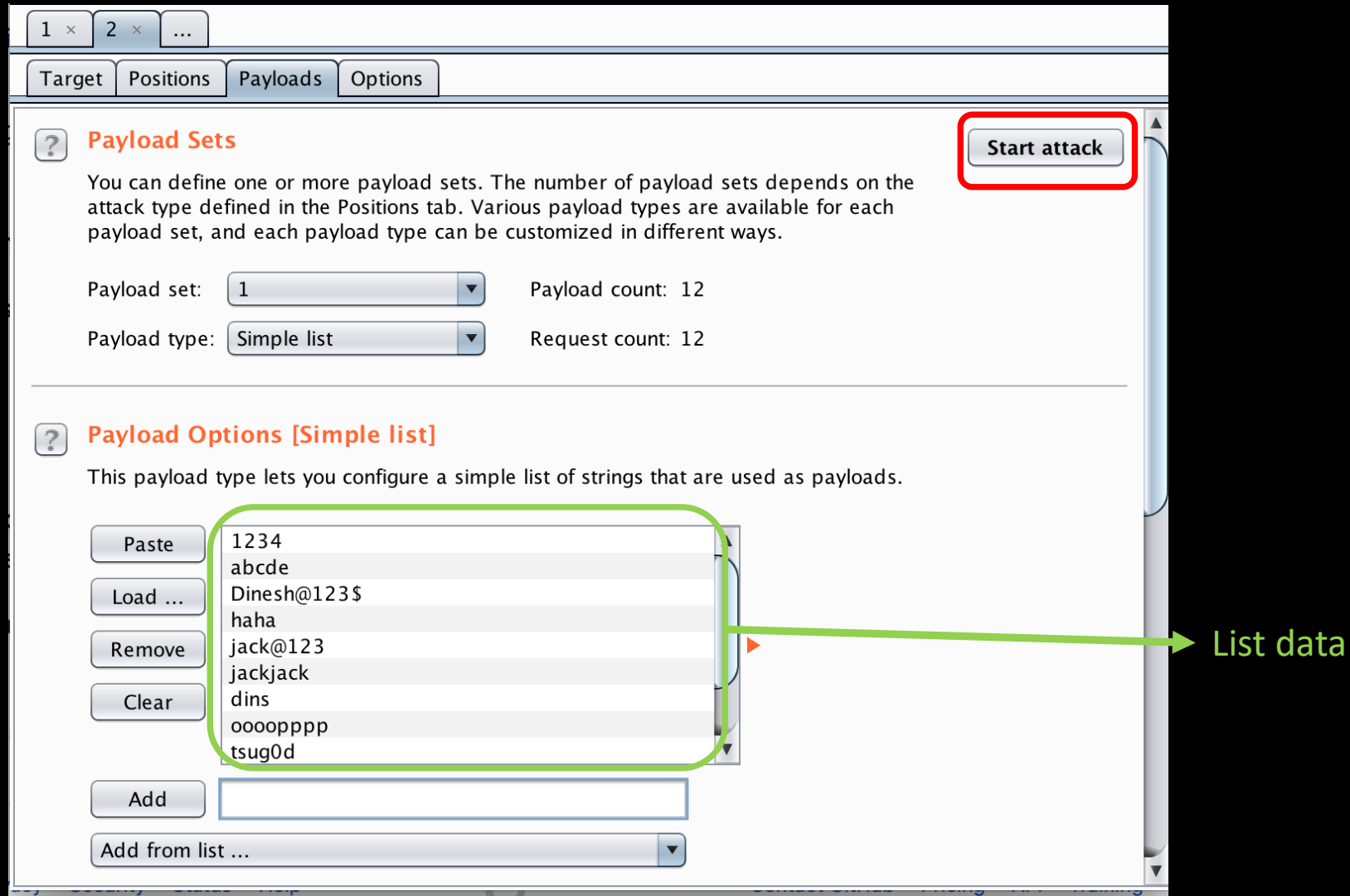
- You can also use **Intruder tab** to perform “brute-force” attack, send the login request to **Intruder**
- In this case, we will perform brute-force password of user “**dinesh**”

The screenshot shows the Burp Suite interface with the **Intruder** tab selected. Below the tab bar, there are buttons for **1 x**, **2 x**, **3 x**, and **...**. The main panel has sub-tabs for **Target**, **Positions**, **Payloads**, and **Options**, with **Positions** currently active. The **Payload Positions** section includes a help icon, a title, and a description: "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." A **Start attack** button is in the top right. The **Attack type** dropdown is set to **Sniper**. The request preview shows an HTTP POST to `/login` with various headers and a body containing `username=dinesh&password=$dkm$`. A green box highlights the `$dkm$` payload, with a green arrow pointing to it and a text label: `$ symbol specify the place to brute-force`. On the right side of the request preview, there are four buttons: **Add \$**, **Clear \$**, **Auto \$**, and **Refresh**.



# More Burp

- Within **intruder**, Go to **Payloads** tab, it's the payload we provide the "list data" to brute, provide it, then **Start attack**



# More Burp

## - Result!

Results Target Positions Payloads Options

Filter: Showing all items

Request	Payload	Status	Error	Timeout	Length	Comment
0		200	<input type="checkbox"/>	<input type="checkbox"/>	201	
1	1234	200	<input type="checkbox"/>	<input type="checkbox"/>	201	
2	abcde	200	<input type="checkbox"/>	<input type="checkbox"/>	201	
3	Dinesh@123\$	200	<input type="checkbox"/>	<input type="checkbox"/>	206	
4	haha	200	<input type="checkbox"/>	<input type="checkbox"/>	201	
5	jack@123	200	<input type="checkbox"/>	<input type="checkbox"/>	201	
6	jackjack	200	<input type="checkbox"/>	<input type="checkbox"/>	201	
7	dins	200	<input type="checkbox"/>	<input type="checkbox"/>	201	
8	oooopppp	200	<input type="checkbox"/>	<input type="checkbox"/>	201	
9	tsug0d	200	<input type="checkbox"/>	<input type="checkbox"/>	201	
10	lalala	200	<input type="checkbox"/>	<input type="checkbox"/>	201	
11	nanannana	200	<input type="checkbox"/>	<input type="checkbox"/>	201	
12	something	200	<input type="checkbox"/>	<input type="checkbox"/>	201	

Request Response

Raw Headers Hex

HTTP/1.1 200 OK  
Content-Type: text/html; charset=utf-8  
Content-Length: 52  
Connection: close  
Date: Tue, 11 Sep 2018 18:55:23 GMT  
Server: localhost

{"message": "Correct Credentials", "user": "dinesh"}

Unique length, so maybe its the correct one!

True!

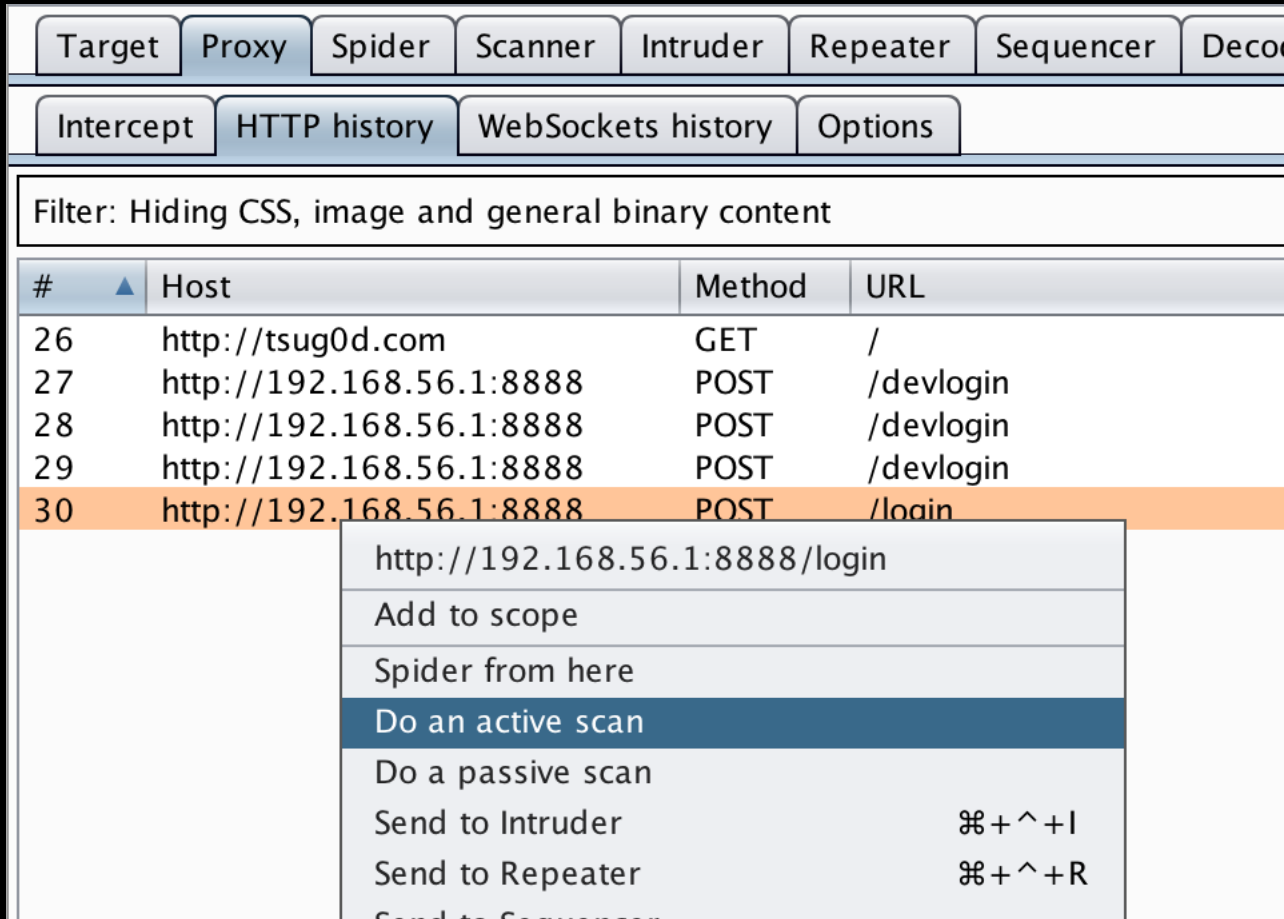
# More Burp

- History, it logs all the request coming from the device

Target	Proxy	Spider	Scanner	Intruder	Repeater	Sequencer	Decoder	Comparer	Extender
Intercept	HTTP history	WebSockets history	Options						
Filter: Hiding CSS, image and general binary content									
#	▲	Host	Method	URL	Params	Edit			
26		http://tsug0d.com	GET	/					
27		http://192.168.56.1:8888	POST	/devlogin	✓				
28		http://192.168.56.1:8888	POST	/devlogin	✓				
29		http://192.168.56.1:8888	POST	/devlogin	✓				
30		http://192.168.56.1:8888	POST	/login	✓				

# More Burp

- Well, its mobile, so we cannot scan? No, we can, just use **Scanner tab** of Burp Suite (Require Burp Suite Professional)



The screenshot shows the Burp Suite interface with the 'Scanner' tab selected. The top navigation bar includes 'Target', 'Proxy', 'Spider', 'Scanner', 'Intruder', 'Repeater', 'Sequencer', and 'Decoder'. Below this, the 'Intercept' tab is active, showing 'HTTP history', 'WebSockets history', and 'Options'. A filter is applied: 'Filter: Hiding CSS, image and general binary content'. The main area displays a table of HTTP requests:

#	Host	Method	URL
26	http://tsug0d.com	GET	/
27	http://192.168.56.1:8888	POST	/devlogin
28	http://192.168.56.1:8888	POST	/devlogin
29	http://192.168.56.1:8888	POST	/devlogin
30	http://192.168.56.1:8888	POST	/login

A context menu is open over the selected request (30), showing the following options:

- http://192.168.56.1:8888/login
- Add to scope
- Spider from here
- Do an active scan**
- Do a passive scan
- Send to Intruder ⌘+^+I
- Send to Repeater ⌘+^+R
- Send to Sequencer

# More Burp

- Come to **Scanner tab**, and collect the result

The screenshot shows the Burp Suite interface with the Scanner tab selected. The main table displays the scan results for a single item.

#	Host	URL	Status	Issues	Requests
1	http://192.168.56.1:8888	/login	finished	3	459

A detailed view of the scan results for item 1 is shown in a pop-up window titled "Scan item 1 | 3 issues | finished | http://192.168.56.1:8888/login". The "Issues" tab is selected, showing the following findings:

- ❗ Cross-site scripting (reflected)
- ? Cross-site request forgery
- i Input returned in response (reflected)

Below the issues, the "Request" tab is selected, showing the raw HTTP request:

```
POST /login HTTP/1.1
Content-Length: 28
Content-Type: application/x-www-form-urlencoded
Host: 192.168.56.1:8888
Connection: close
User-Agent: Apache-HttpClient/UNAVAILABLE (java 1.4)

username=dineshb4ekj%3cscript%3ealert(1)%3c%2fscript%3esipog&password=dkm
```

The raw request shows a POST to /login with a Content-Type of application/x-www-form-urlencoded. The request body contains a payload designed to trigger a cross-site scripting (XSS) alert.