STI

Sécurité des Technologies Internet

Laboratoire

bWAPP & Burp Suite

Professeur	Assistant	Assistante
Abraham Rubinstein	Yann Lederrey	Lucie Steiner
abraham.rubinstein@heig-vd.ch	yann.lederrey@heig-vd.ch	lucie.steiner@heig-vd.ch

septembre 2019 – février 2020

Table des matières

1. bWAPP

bWAPP is a buggy and deliberately insecure web application. Its goal is to help anyone who want to discover and prevent web vulnerabilities. Currently, bWAPP offers over 100 different web vulnerabilities that can be experienced directly in your browser. Developed in PHP, bWAPP use MySQL as database and can be hosted on Linux, Windows or Mac OSX (WAMP, LAMP or XAMPP).

You can either download the web application source code or a complete virtualized system named **bee-box** (Linux VM with bWAPP pre-installed). You can also use a Docker image. To avoid unnecessary deployment/installation issues, we are going to detail the use of a Docker image and also the bee-box all-in-one solution. All you need to do is to download the VM archive and import it into your favorite virtualization software.

For the entire lab, we recommend the use of the following sites:

- bWAPP website
  http://www.itsecgames.com/
- bWAPP web application
  http://sourceforge.net/projects/bwapp/files/bWAPP/
- Docker image
  https://hub.docker.com/r/raesene/bwapp
- Bee-box VM
  http://sourceforge.net/projects/bwapp/files/bee-box/

1.1. Installation

As already said, you can either manually install bWAPP on your computer or use the ready-to-use Docker image or VM files. For this lab, we are going to use the provided VM or the Docker solution but the manual install way will be explained as well.

1.1.1. Manual Installation

For a manual installation, you need to satisfy the following requirements:

- **Operating system**
  Windows, Linux, Unix, Mac OS, …
- **Web server**
  Apache, Nginx, IIS, etc with PHP extensions
- **Database**
  MySQL

Instead of managing each part separately, you can use a package like WAMP or XAMPP. We will not describe web server installation and configuration here. We assume that people who want to install it manually (instead of using a VM) have the necessary knowledge to set up their own web server. Please note that it requires lot of configuration (Apache, PHP, Postfix, etc) because of the number of different vulnerabilities available.

Once the web server is ready and running, download the bWAPP archive from the URL above and deploy bWAPP web application:

1. Extract the zip file
   $ unzip bWAPPv2.2.zip
2. Move the directory “bWAPP” to the root of your web server
   $ cp –r bWAPP /var/www/

3. Give full permission to the directories “passwords”, “images”, “documents” and “logs”
   $ chmod 777 passwords/
   $ chmod 777 images/
   $ chmod 777 documents/
   $ chmod 777 logs/
4. Edit the file “admin/settings.php” with your own database connection settings
5. Open the file “install.php” in your browser
   https://localhost/bWAPP/install.php
6. Go to the login page and use the default credentials **bee/bug**
   https://localhost/bWAPP/login.php
7. You are all set up and ready to explore bWAPP !

1.1.2. Docker Image

Several Docker images are available. Because this is a web-based application, Docker is a perfect solution for bWAPP.

A lot of the work on bWAPP implies the use of an intercepting proxy. Working with this kind of tool will become much easier when your webapp runs on port 80. So, you will need to grant administrator privileges when running your Docker container so that it can bind to port 80.

The following steps will get you ready shortly:

1. Download the image and run a container binding port 80
   $ docker run -d -p 80:80 raesene/bwapp
2. Browse to http://localhost/install.php and click on the **here** link in order to prepare the database
3. Login with default bWAPP credentials **bee/bug**
4. You are all setup and ready to explore bWAPP

1.1.3. Virtual Machine

The **bee-box** archive contains the virtual machine in VMDK format. You can use VMware as well as VirtualBox to load and run it. The following steps will get you ready shortly:

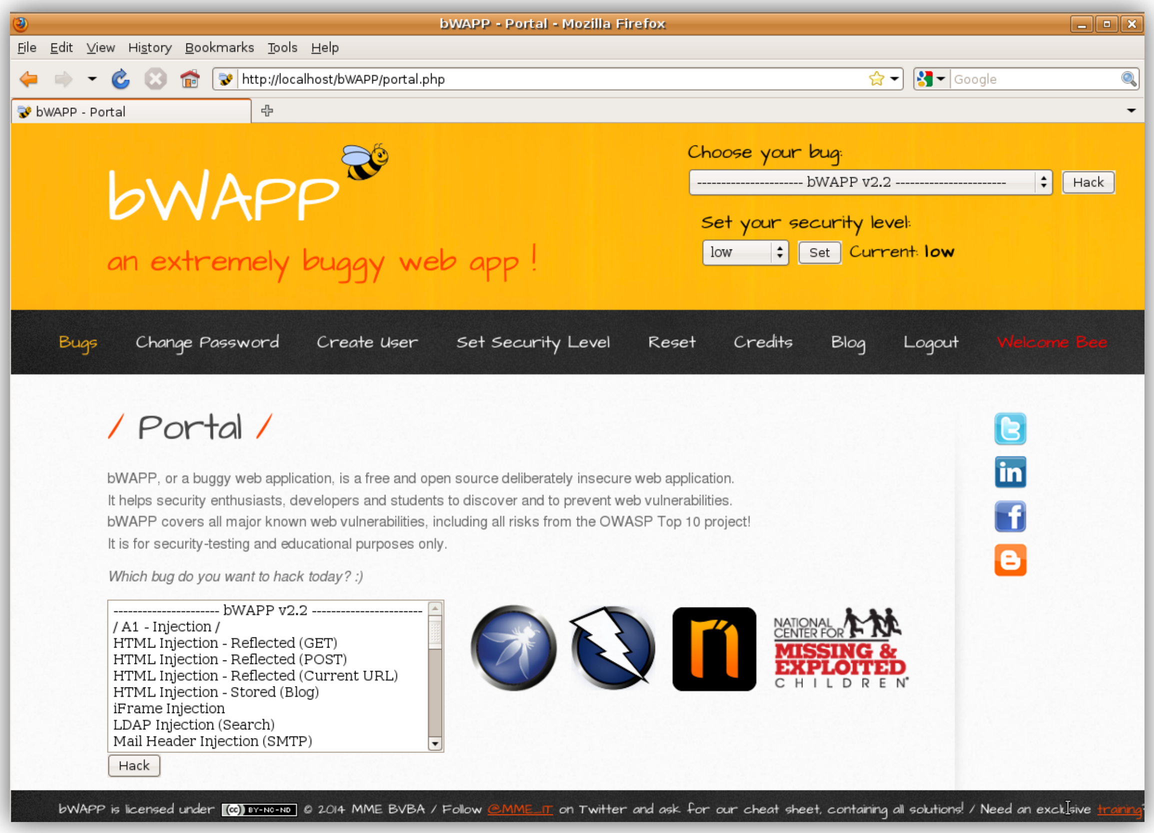
1. Extract the archive
   $ 7z e bee-box_v1.6.7z
2. Load the VM
   Under VMware, import the virtual machine by simply double clicking on “bee-box.vmx”. With VirtualBox, you need to create a new virtual machine “Ubuntu 64bits” and select “bee-box.vmdk” as hard drive.
3. Configure the network
   Ensure that your VM network card is correctly configured with NAT.
4. Start the VM
5. Browse to http://localhost/bWAPP/login.php and login with default bWAPP credentials **bee/bug**
6. You are all setup and ready to explore bWAPP

Usually, you don’t need to alter the virtual machine. But, if you want to make some modification, here are the default credentials:

• Linux credentials	bee/bug root/bug
• Mysql credentials	root/bug

It's a good habit to take a snapshot of the VM before hacking the bee-box. Thus, you will be able to return in a clean state if you do anything wrong. Also, don't upgrade the Linux operating system. Some package contains vulnerabilities that will be patched if you do so.

1.2. How To Use It


bWAPP - Portal - Mozilla Firefox
File Edit View History Bookmarks Tools Help
http://localhost/bWAPP/portal.php
Google
bWAPP - Portal
bWAPP
an extremely buggy web app !
Choose your bug:
---------------------- bWAPP v2.2 ----------------------
Hack
Set your security level:
low
Set
Current: low
Bugs
Change Password
Create User
Set Security Level
Reset
Credits
Blog
Logout
Welcome Bee
Hack
NATIONAL CENTER FOR MISSING & EXPLOITED CHILDREN

/ Portal /

bWAPP, or a buggy web application, is a free and open source deliberately insecure web application.
It helps security enthusiasts, developers and students to discover and to prevent web vulnerabilities.
bWAPP covers all major known web vulnerabilities, including all risks from the OWASP Top 10 project!
It is for security-testing and educational purposes only.

Which bug do you want to hack today? :)

```
---------------------- bWAPP v2.2 ----------------------
/ A1 - Injection /
HTML Injection - Reflected (GET)
HTML Injection - Reflected (POST)
HTML Injection - Reflected (Current URL)
HTML Injection - Stored (Blog)
iFrame Injection
LDAP Injection (Search)
Mail Header Injection (SMTP)
```

bWAPP is licensed under (cc) BY-NC-ND © 2014 MME BVBA / Follow @MME_IT on Twitter and ask for our cheat sheet, containing all solutions! / Need an exclusive training

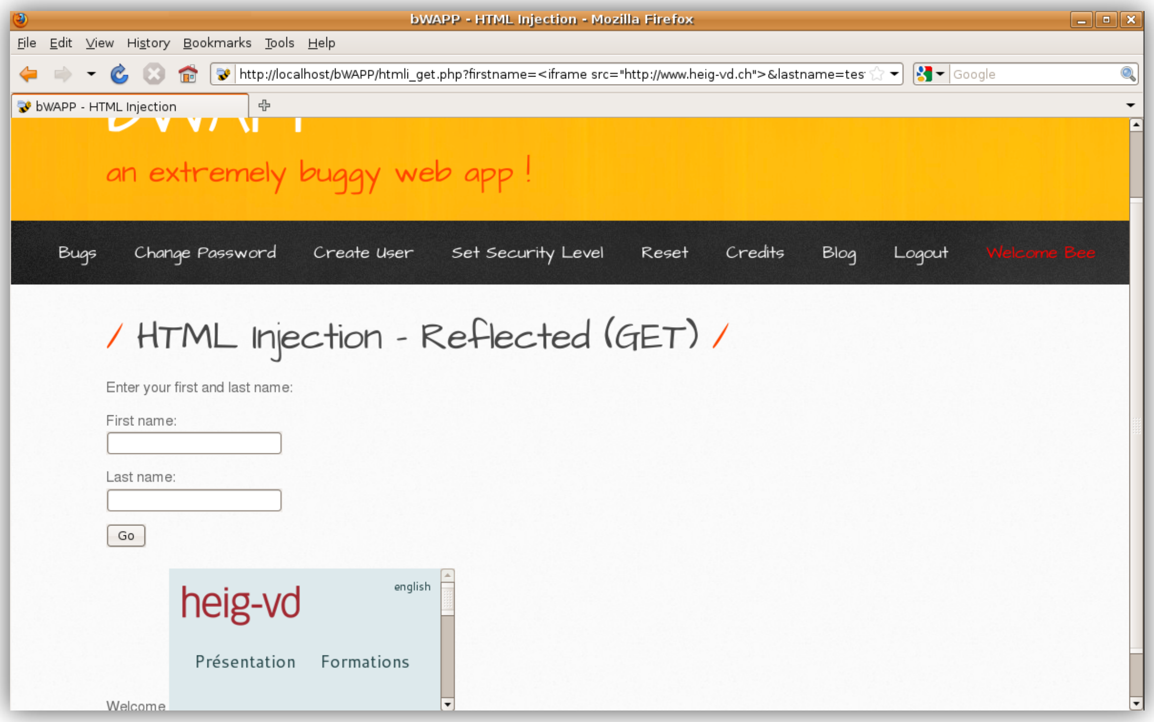
Figure 1 : bWAPP portal

bWAPP is really simple to use. At the top right, we can choose which attack we want to discover and the level of security we want to apply. It is recommended to use the low security level since medium or high level will disable some attacks.

For example, we select the first one named "*HTML Injection – Reflected (GET)*" and click "*Hack*". Now, the page is displaying a HTML form asking for our first and last name. We can test it with random data and see that it is doing a GET request, printing back our inputs.

How to attack it? Basically, it's called "HTML Injection", so we can try to inject any HTML code into both fields of the form. We will see that no input validation is made, nor client-side, nor server-side.

For example, we can inject an iframe like this.


bWAPP - HTML Injection - Mozilla Firefox
File Edit View History Bookmarks Tools Help
bWAPP - HTML Injection
an extremely buggy web app !
Bugs Change Password Create User Set Security Level Reset Credits Blog Logout Welcome Bee
Go
heig-vd
english
Présentation Formations

http://localhost/bWAPP/htmli_get.php?firstname=<iframe src="http://www.heig-vd.ch">&lastname=tes

/ HTML Injection - Reflected (GET) /

Enter your first and last name:

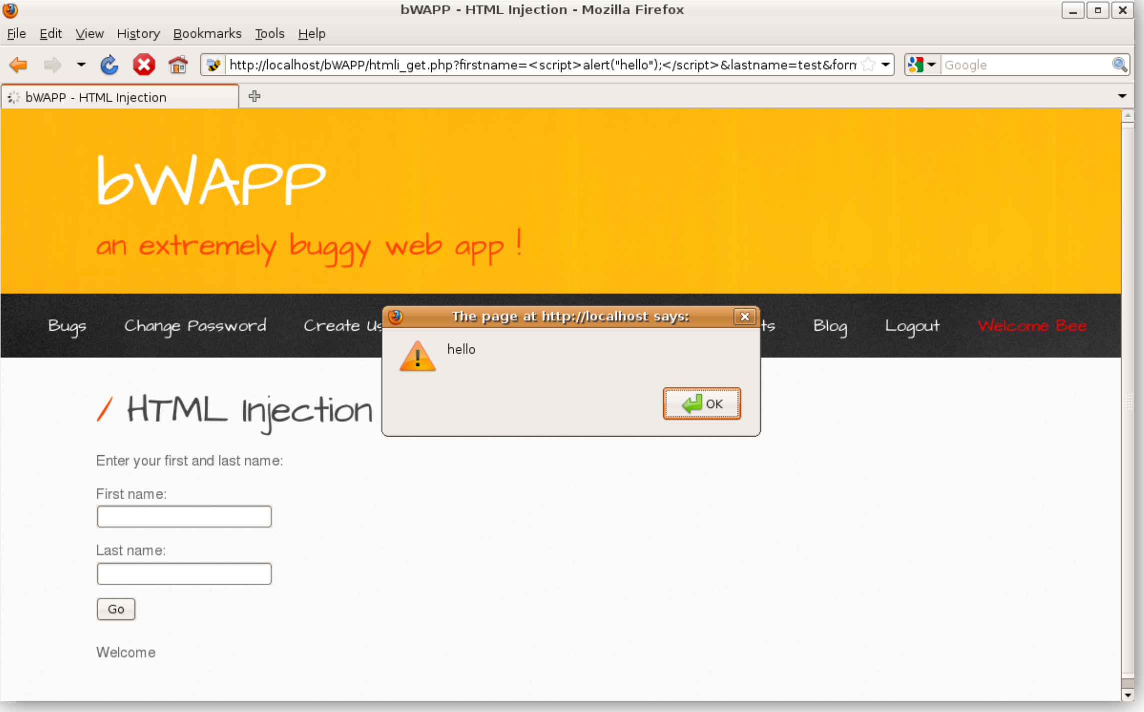
First name:

Last name:

Welcome

Figure 2 : HTML Injection – Reflected (GET)

The iframe is directly embed in the page. More interestingly, we are able to inject Javascript with the HTML script tag.


bWAPP - HTML Injection - Mozilla Firefox
File Edit View History Bookmarks Tools Help
bWAPP - HTML Injection
bWAPP
an extremely buggy web app !
Bugs Change Password Create Us ts Blog Logout Welcome Bee
OK
Go

http://localhost/bWAPP/htmli_get.php?firstname=<script>alert("hello");</script>&lastname=test&forn

The page at http://localhost says:

hello

/ HTML Injection

Enter your first and last name:

First name:

Last name:

Welcome

Figure 3 : HTML and Javascript injection

With Javascript, we can do more than just showing an alert. The common use is to make a cookie stealer (XSS) or alter any html part of the page (phishing). Then, we could send our malicious crafted URL to fool our victim.

1.3. Next Steps

bWAPP is a powerful tool to experiment lot of different web-based vulnerabilities. First, you must learn attacks theory by googling, reading books, etc and then experiment them directly on a real website (bWAPP). However, there's many different way to exploit bWAPP. Thus, it will never told you if you are doing things right or if your exploitation is a success or not. It is up to you to exploit security hole the way you want and stop when you think you have achieved your goal.

For example, regarding the HTML injection attack, you could be satisfied to print a Javascript alert box where someone else will stop only after having achieved a real and working XSS.

Bee-box VM include the browser Firefox which contains a set of add-ons that will certainly fit your needs during your experimentation. Also, in the next chapter, we are going to talk about a well-known security tool oriented web application called Burp Suite. This latter could be easily downloaded directly inside the VM and used against bWAPP.

2. Burp Suite

Burp Suite is an integrated platform for performing security testing of web applications. It contains various tools letting you combine advanced manual techniques with state-of-the-art automation, to make your work faster, more effective, and more fun. The following key components are:

- **Proxy**
  As a proxy, you are able to intercept, inspect and modify traffic between your browser and the target application.
- **Spider**
  Wep application crawler for mapping website. Allows you to see the content and functionality of a web application.
- **Scanner**
  Perform advanced scanning for automating the detection of numerous types of well-known vulnerabilities.
- **Intruder**
  Used for performing powerful and customized attacks.
- **Repeater**
  Modifying and reissuing individual HTTP requests, and analyzing their responses.
- **Sequencer**
  Analyzing the degree of randomness in security-critical tokens issued by an application.

In the context of this lab, we will mainly use the proxy feature of Burp Suite against bWAPP. When you need to alter headers or POST data, for example,, Burp Suite become really helpful.

Some interesting link to get you started:

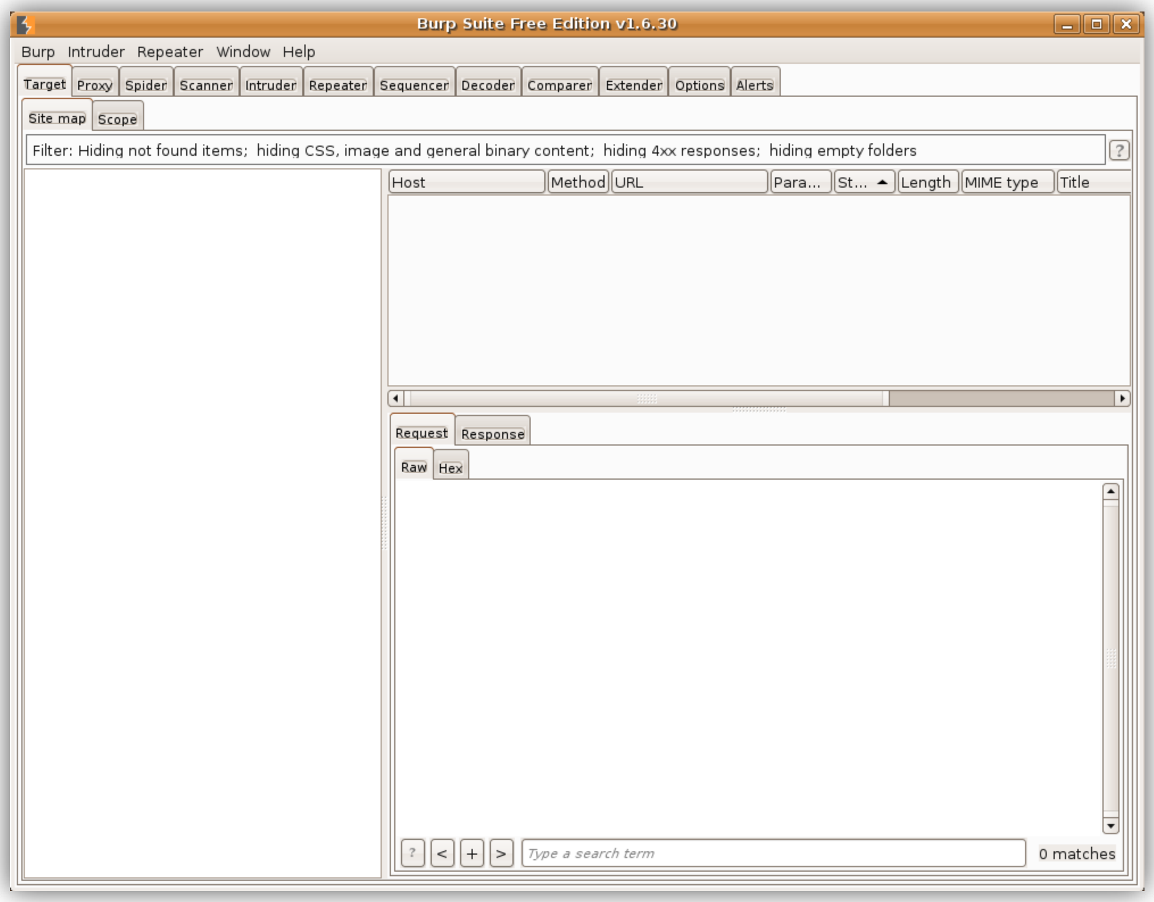
- Burp Website
  https://portswigger.net
- Free Edition
  https://portswigger.net/burp/downloadfree.html
- Official Documentation
  https://portswigger.net/burp/help/

2.1. Installation

Burp doesn't need to be installed. It is distributed as a simple jar file which you can run using official Java VM or open-source version, OpendJDK.

1. Download it from the link above
2. Run it
   $ java –jar burpsuite_free_v1.6.25.jar

2.2. How To Use It


Burp Suite Free Edition v1.6.30
Burp Intruder Repeater Window Help
Target Proxy Spider Scanner Intruder Repeater Sequencer Decoder Comparer Extender Options Alerts
Site map Scope
Host Method URL Para... St... Length MIME type Title
Request Response
Raw Hex
? < + >
Type a search term
0 matches

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

Figure 4 : Burp Suite interface

Burp Suite is a complex tool and contains many advanced features. In the context of our lab, we will study only the proxy component. Feel free to investigate the official documentation to learn how to use the other components.

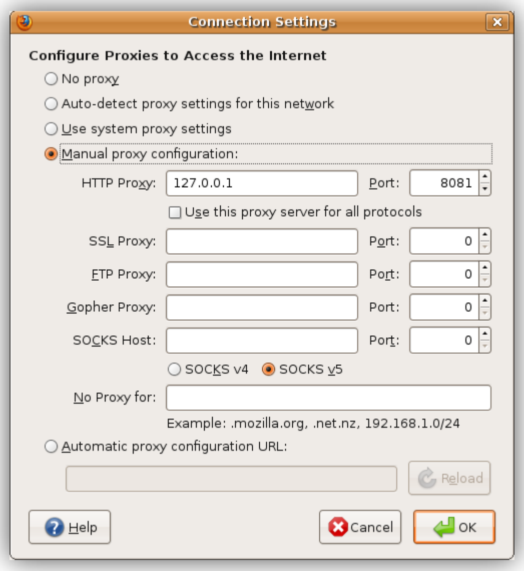
At first, we need to enable the proxy:

1. Go to the **Proxy** > **Options** tab
2. **Edit** the default proxy (127.0.0.1.8080) and change the port to **8081**
   For unknown reason, it appears that the default port (8080) cannot be listening to.
3. **Check** the "running" box of the only entry to start the proxy

Now that the proxy is running, we must configure our browser to take it into account.

1. Open Firefox and go to **Edit** > **Preferences**
2. Under **Advanced** tab, go to **Network** then **Settings** button
3. Check "Manual proxy configuration"
   a. Set "HTTP Proxy" to **127.0.0.1**
   b. Set the "Port" to **8081**
   c. Clear the "No Proxy for" field

Our configuration looks like this:



Connection Settings

Configure Proxies to Access the Internet

○ No proxy
○ Auto-detect proxy settings for this network
○ Use system proxy settings
● Manual proxy configuration:

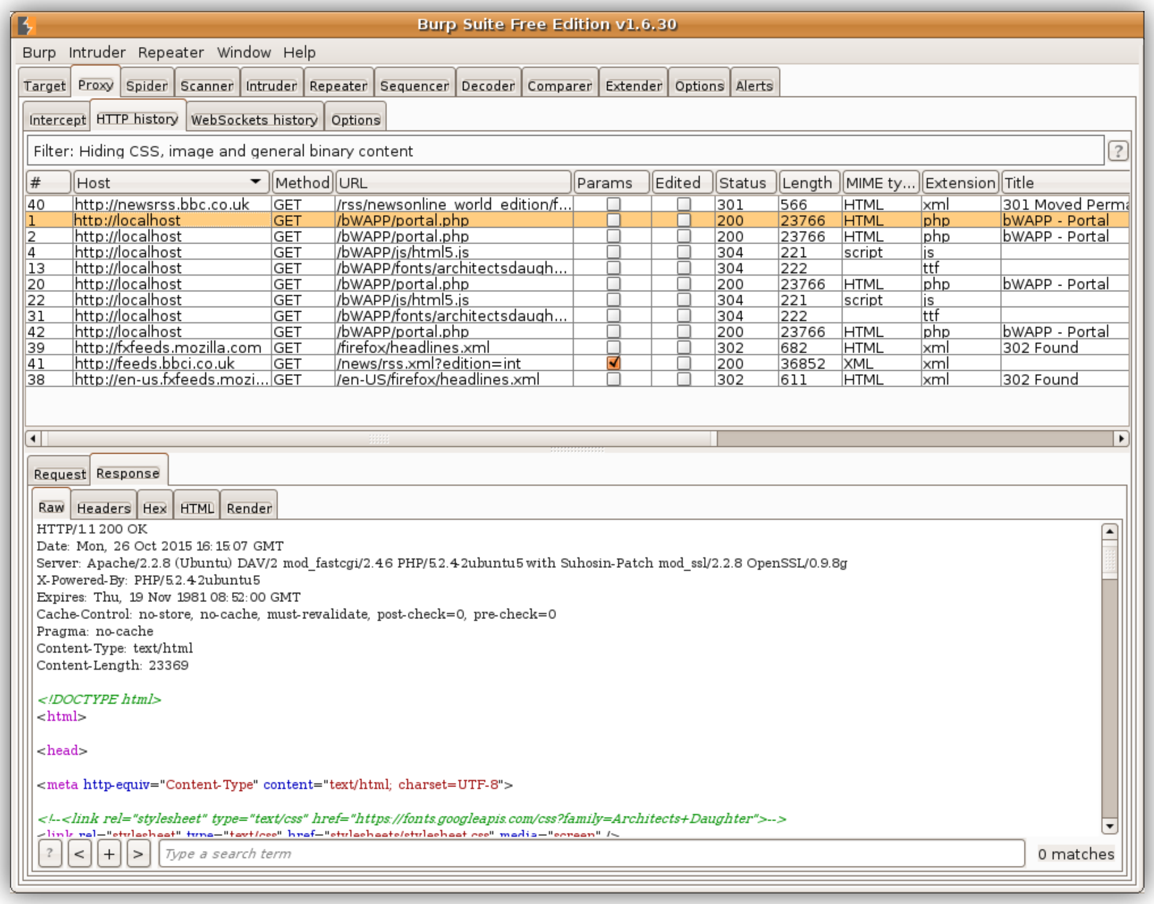
HTTP Proxy: 127.0.0.1 Port: 8081
□ Use this proxy server for all protocols
SSL Proxy: Port: 0
FTP Proxy: Port: 0
Gopher Proxy: Port: 0
SOCKS Host: Port: 0
○ SOCKS v4 ● SOCKS v5
No Proxy for:
Example: .mozilla.org, .net.nz, 192.168.1.0/24

○ Automatic proxy configuration URL:

Figure 5 : Firefox proxy settings

In Burp Suite, we ensure that the interceptor is disabled in **Proxy** > **Intercept**. The third button should be "Intercept is off". If not, just click on it to disable it.

Now, we can use Firefox to navigate on bWAPP or Internet and we will see all our HTTP traffics logged in **Proxy** > **HTTP history**.



Burp Suite Free Edition v1.6.30
Burp Intruder Repeater Window Help
Target Proxy Spider Scanner Intruder Repeater Sequencer Decoder Comparer Extender Options Alerts
Intercept HTTP history WebSockets history Options

Filter: Hiding CSS, image and general binary content

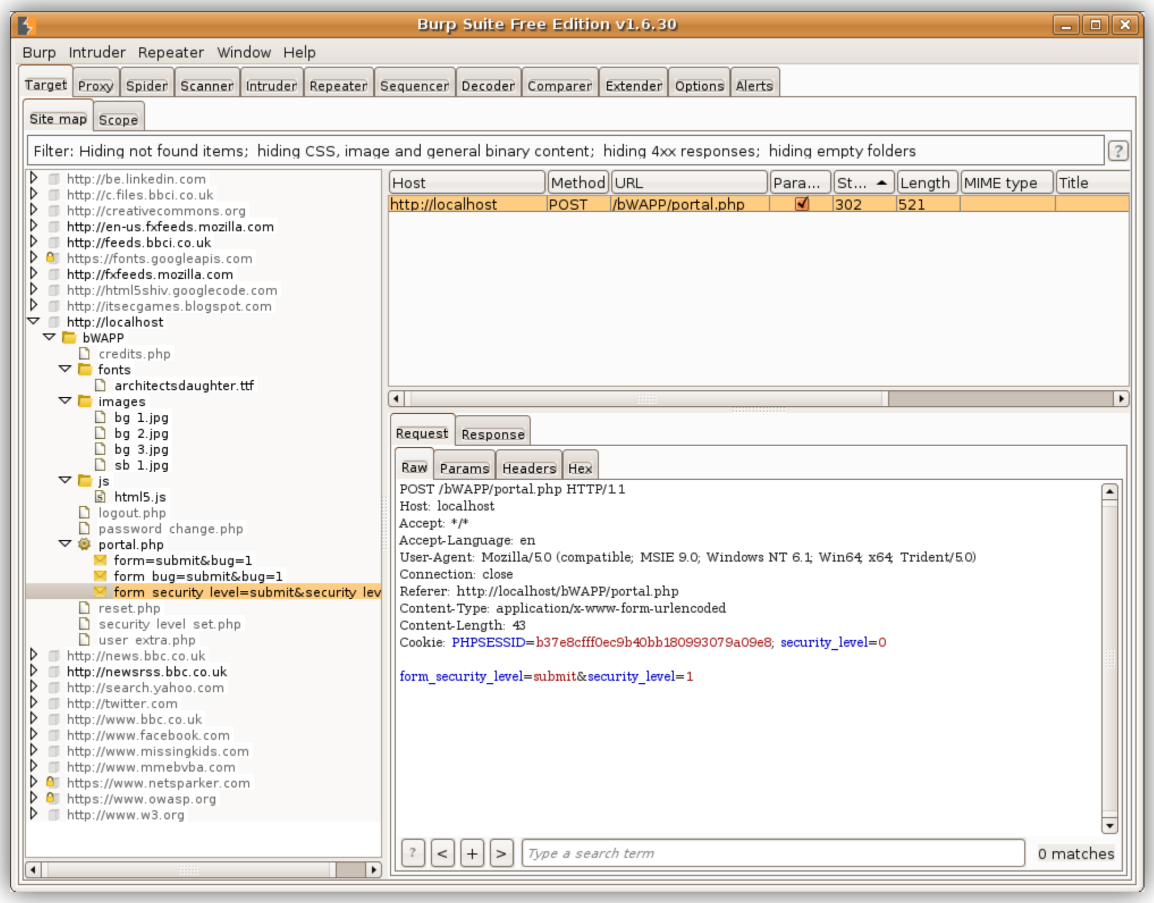
#	Host	Method	URL	Params	Edited	Status	Length	MIME ty...	Extension	Title
40	http://newsrss.bbc.co.uk	GET	/rss/newsonline_world_edition/f...	□	□	301	566	HTML	xml	301 Moved Perma
1	http://localhost	GET	/bWAPP/portal.php	□	□	200	23766	HTML	php	bWAPP - Portal
2	http://localhost	GET	/bWAPP/portal.php	□	□	200	23766	HTML	php	bWAPP - Portal
4	http://localhost	GET	/bWAPP/js/html5.js	□	□	304	221	script	js	
13	http://localhost	GET	/bWAPP/fonts/architectsdaugh...	□	□	304	222		ttf	
20	http://localhost	GET	/bWAPP/portal.php	□	□	200	23766	HTML	php	bWAPP - Portal
22	http://localhost	GET	/bWAPP/js/html5.js	□	□	304	221	script	js	
31	http://localhost	GET	/bWAPP/fonts/architectsdaugh...	□	□	304	222		ttf	
42	http://localhost	GET	/bWAPP/portal.php	□	□	200	23766	HTML	php	bWAPP - Portal
39	http://fxfeeds.mozilla.com	GET	/firefox/headlines.xml	□	□	302	682	HTML	xml	302 Found
41	http://feeds.bbci.co.uk	GET	/news/rss.xml?edition=int	☑	□	200	36852	XML	xml	
38	http://en-us.fxfeeds.mozi...	GET	/en-US/firefox/headlines.xml	□	□	302	611	HTML	xml	302 Found

```
HTTP/1.1 200 OK
Date: Mon, 26 Oct 2015 16:15:07 GMT
Server: Apache/2.2.8 (Ubuntu) DAV/2 mod_fastcgi/2.4.6 PHP/5.2.4-2ubuntu5 with Suhosin-Patch mod_ssl/2.2.8 OpenSSL/0.9.8g
X-Powered-By: PHP/5.2.4-2ubuntu5
Expires: Thu, 19 Nov 1981 08:52:00 GMT
Cache-Control: no-store, no-cache, must-revalidate, post-check=0, pre-check=0
Pragma: no-cache
Content-Type: text/html
Content-Length: 23369

<!DOCTYPE html>
<html>

<head>

<meta http-equiv="Content-Type" content="text/html; charset=UTF-8">

<!--<link rel="stylesheet" type="text/css" href="https://fonts.googleapis.com/css?family=Architects+Daughter">-->
<link rel="stylesheet" type="text/css" href="stylesheets/stylesheet.css" media="screen" />
```

Figure 6 : Burp Suite history HTTP requests

We can watch each request and response as well as in different format (raw, hex, html, etc). From there, we can select some request to send to other Burp component like Repeater, Sequencer, etc.

Previously, we have briefly mentioned Spider. It automatically runs in background and analyzes the HTTP traffic to try to recover information about the web application mapping. In **Target** > **Site map** tab, Spider gives us a detailed view of the bWAPP website structure based on our browsing.


Burp Suite Free Edition v1.6.30
Burp Intruder Repeater Window Help
Target Proxy Spider Scanner Intruder Repeater Sequencer Decoder Comparer Extender Options Alerts
Site map Scope
Request Response
Raw Params Headers Hex
? < + >
Type a search term
0 matches

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

```
http://be.linkedin.com
http://c.files.bbci.co.uk
http://creativecommons.org
http://en-us.fxfeeds.mozilla.com
http://feeds.bbci.co.uk
https://fonts.googleapis.com
http://fxfeeds.mozilla.com
http://html5shiv.googlecode.com
http://itsecgames.blogspot.com
http://localhost
  bWAPP
    credits.php
    fonts
      architectsdaughter.ttf
    images
      bg 1.jpg
      bg 2.jpg
      bg 3.jpg
      sb 1.jpg
    js
      html5.js
    logout.php
    password change.php
    portal.php
      form=submit&bug=1
      form bug=submit&bug=1
      form security level=submit&security lev
    reset.php
    security level set.php
    user extra.php
http://news.bbc.co.uk
http://newsrss.bbc.co.uk
http://search.yahoo.com
http://twitter.com
http://www.bbc.co.uk
http://www.facebook.com
http://www.missingkids.com
http://www.mmebvba.com
https://www.netsparker.com
https://www.owasp.org
http://www.w3.org
```

Host	Method	URL	Para...	St... ▲	Length	MIME type	Title
http://localhost	POST	/bWAPP/portal.php	☑	302	521		

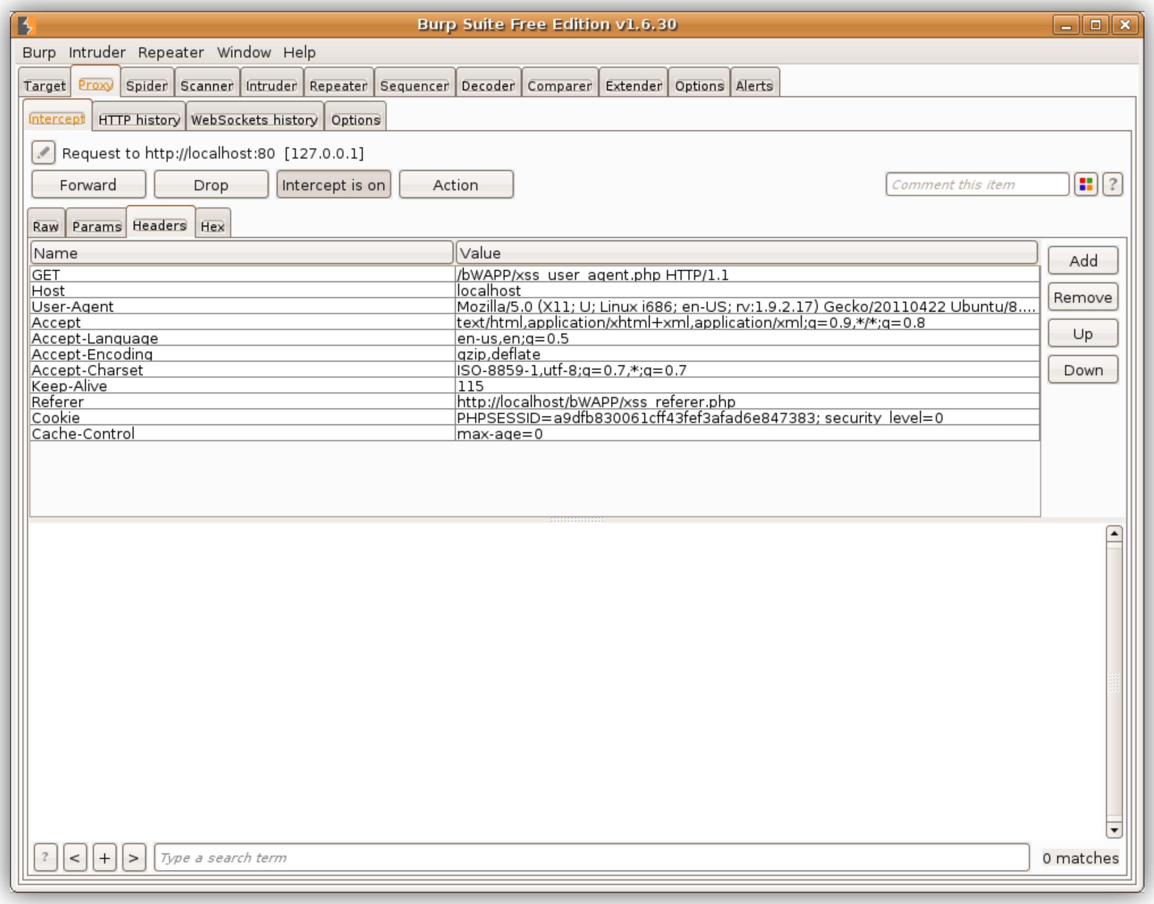
```
POST /bWAPP/portal.php HTTP/1.1
Host: localhost
Accept: */*
Accept-Language: en
User-Agent: Mozilla/5.0 (compatible; MSIE 9.0; Windows NT 6.1; Win64; x64; Trident/5.0)
Connection: close
Referer: http://localhost/bWAPP/portal.php
Content-Type: application/x-www-form-urlencoded
Content-Length: 43
Cookie: PHPSESSID=b37e8cfff0ec9b40bb180993079a09e8; security_level=0

form_security_level=submit&security_level=1
```

Figure 7 : Burp Suite Spider

1.1.4. Concrete Case

Using bWAPP, we are going to show a real case using Burp Suite and its proxy feature. We chose to attack the "*XSS - Reflected (User-Agent)*" vulnerability.

First, we go to Burp Suite in **Proxy** > **Intercept** and set intercept to "Intercept is on". In Firefox, we reload the attack page and see that the request is pending. Back to Burp Suite, our request have been correctly intercepted. We can modify it the way we want before forwarding it.

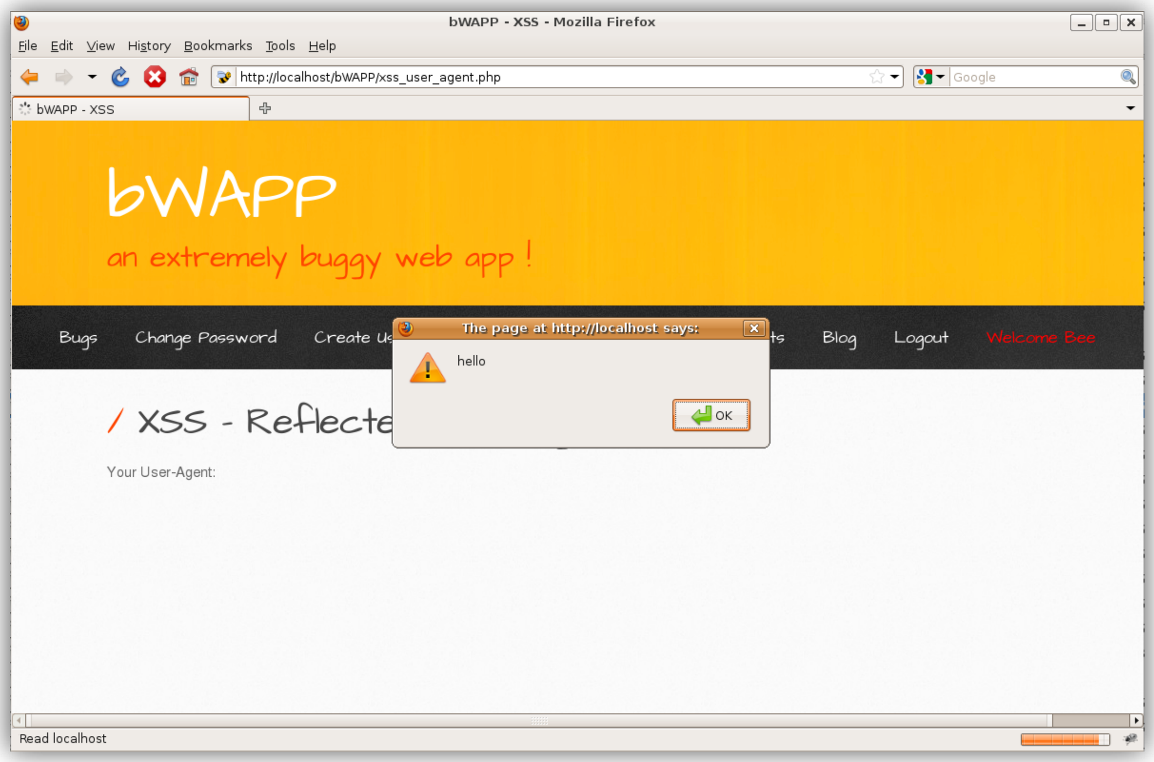

Burp Suite Free Edition v1.6.30
Burp Intruder Repeater Window Help
Target Proxy Spider Scanner Intruder Repeater Sequencer Decoder Comparer Extender Options Alerts
Intercept HTTP history WebSockets history Options
Forward Drop Intercept is on Action
Comment this item
Raw Params Headers Hex
Add Remove Up Down
Type a search term
0 matches

Request to http://localhost:80 [127.0.0.1]

Name	Value
GET	/bWAPP/xss_user_agent.php HTTP/1.1
Host	localhost
User-Agent	Mozilla/5.0 (X11; U; Linux i686; en-US; rv:1.9.2.17) Gecko/20110422 Ubuntu/8....
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
Referer	http://localhost/bWAPP/xss_referer.php
Cookie	PHPSESSID=a9dfb830061cff43fef3afad6e847383; security_level=0
Cache-Control	max-age=0

Figure 8 : Burp Suite HTTP interception

As the vulnerability's title suggest, it is possible to inject code into the header "User-Agent". Thanks to Burp Suite, we can alter the request and place some HTML and javascript inside. For the demo, we simply inject a Javascrip alert and forward the request.


bWAPP - XSS - Mozilla Firefox
File Edit View History Bookmarks Tools Help
http://localhost/bWAPP/xss_user_agent.php
Google
bWAPP - XSS
bWAPP
an extremely buggy web app !
Bugs
Change Password
Create Us
ts
Blog
Logout
Welcome Bee
/ XSS - Reflecte
OK
Read localhost

The page at http://localhost says:

hello

Your User-Agent:

Figure 9 : Header HTML/Javascript injection

As we can see, the Javascript have been successfully executed.

3. Available labs

bWAPP offers the following exercises:

1. Injection
    - HTML Injection - Reflected (GET)
    - HTML Injection - Reflected (POST)
    - HTML Injection - Reflected (Current URL)
    - HTML Injection - Stored (Blog)
    - iFrame Injection
    - LDAP Injection (Search)
    - Mail Header Injection (SMTP)
    - OS Command Injection
    - OS Command Injection - Blind
    - PHP Code Injection
    - Server-Side Includes (SSI) Injection
    - SQL Injection (GET/Search)
    - SQL Injection (GET/Select)
    - SQL Injection (POST/Search)
    - SQL Injection (POST/Select)
    - SQL Injection (AJAX/JSON/jQuery)
    - SQL Injection (CAPTCHA)
    - SQL Injection (Login Form/Hero)
    - SQL Injection (Login Form/User)
    - SQL Injection (SQLite)
    - SQL Injection (Drupal)
    - SQL Injection - Stored (Blog)
    - SQL Injection - Stored (SQLite)
    - SQL Injection - Stored (User-Agent)
    - SQL Injection - Stored (XML)
    - SQL Injection - Blind - Boolean-Based
    - SQL Injection - Blind - Time-Based
    - SQL Injection - Blind (SQLite)
    - SQL Injection - Blind (Web Services/SOAP)
    - XML/XPath Injection (Login Form)
    - XML/XPath Injection (Search)
2. Broken Auth. & Session Mgmt
    - Broken Authentication - CAPTCHA Bypassing
    - Broken Authentication - Forgotten Function
    - Broken Authentication - Insecure Login Forms
    - Broken Authentication - Logout Management
    - Broken Authentication - Password Attacks
    - Broken Authentication - Weak Passwords
    - Session Management - Administrative Portals
    - Session Management - Cookies (HTTPOnly)
    - Session Management - Cookies (Secure)
    - Session Management - Session ID in URL
    - Session Management - Strong Sessions
3. Cross-Site Scripting (XSS)
    - Cross-Site Scripting - Reflected (GET)
    - Cross-Site Scripting - Reflected (POST)
    - Cross-Site Scripting - Reflected (JSON)
    - Cross-Site Scripting - Reflected (AJAX/JSON)

- Cross-Site Scripting - Reflected (AJAX/XML)
- Cross-Site Scripting - Reflected (Back Button)
- Cross-Site Scripting - Reflected (Custom Header)
- Cross-Site Scripting - Reflected (Eval)
- Cross-Site Scripting - Reflected (HREF)
- Cross-Site Scripting - Reflected (Login Form)
- Cross-Site Scripting - Reflected (phpMyAdmin)
- Cross-Site Scripting - Reflected (PHP_SELF)
- Cross-Site Scripting - Reflected (Referer)
- Cross-Site Scripting - Reflected (User-Agent)
- Cross-Site Scripting - Stored (Blog)
- Cross-Site Scripting - Stored (Change Secret)
- Cross-Site Scripting - Stored (Cookies)
- Cross-Site Scripting - Stored (SQLiteManager)
- Cross-Site Scripting - Stored (User-Agent)

4. Insecure Direct Object References
   - Insecure DOR (Change Secret)
   - Insecure DOR (Reset Secret)
   - Insecure DOR (Order Tickets)
5. Security Misconfiguration
   - Arbitrary File Access (Samba)
   - Cross-Domain Policy File (Flash)
   - Cross-Origin Resource Sharing (AJAX)
   - Cross-Site Tracing (XST)
   - Denial-of-Service (Large Chunk Size)
   - Denial-of-Service (Slow HTTP DoS)
   - Denial-of-Service (SSL-Exhaustion)
   - Denial-of-Service (XML Bomb)
   - Insecure FTP Configuration
   - Insecure SNMP Configuration
   - Insecure WebDAV Configuration
   - Local Privilege Escalation (sendpage)
   - Local Privilege Escalation (udev)
   - Man-in-the-Middle Attack (HTTP)
   - Man-in-the-Middle Attack (SMTP)
   - Old/Backup & Unreferenced Files
   - Robots File
6. Sensitive Data Exposure
   - Base64 Encoding (Secret)
   - BEAST/CRIME/BREACH Attacks
   - Clear Text HTTP (Credentials)
   - Heartbleed Vulnerability
   - Host Header Attack (Reset Poisoning)
   - HTML5 Web Storage (Secret)
   - POODLE Vulnerability
   - SSL 2.0 Deprecated Protocol
   - Text Files (Accounts)
7. Missing Functional Level Access Control
   - Directory Traversal - Directories
   - Directory Traversal - Files
   - Host Header Attack (Cache Poisoning)
   - Host Header Attack (Reset Poisoning)

- Local File Inclusion (SQLiteManager)
- Remote & Local File Inclusion (RFI/LFI)
- Restrict Device Access
- Restrict Folder Access
- Server Side Request Forgery (SSRF)
- XML External Entity Attacks (XXE)

8. Cross-Site Request Forgery (CSRF)
    - Cross-Site Request Forgery (Change Password)
    - Cross-Site Request Forgery (Change Secret)
    - Cross-Site Request Forgery (Transfer Amount)
9. Using Known Vulnerable Components
    - Buffer Overflow (Local)
    - Buffer Overflow (Remote)
    - Drupal SQL Injection (Drupageddon)
    - Heartbleed Vulnerability
    - PHP CGI Remote Code Execution
    - PHP Eval Function
    - phpMyAdmin BBCode Tag XSS
    - Shellshock Vulnerability (CGI)
    - SQLiteManager Local File Inclusion
    - SQLiteManager PHP Code Injection
    - SQLiteManager XSS
10. Unvalidated Redirects & Forwards
    - Unvalidated Redirects & Forwards (1)
    - Unvalidated Redirects & Forwards (2)
11. Other bugs…
    - ClickJacking (Movie Tickets)
    - Client-Side Validation (Password)
    - HTTP Parameter Pollution
    - HTTP Response Splitting
    - HTTP Verb Tampering
    - Information Disclosure - Favicon
    - Information Disclosure - Headers
    - Information Disclosure - PHP version
    - Information Disclosure - Robots File
    - Insecure iFrame (Login Form)
    - Unrestricted File Upload