iGNITE Technologies

Burp Suite for Pentester Active Scan++

WWW.HACKINGARTICLES.IN

Contents

Introduction

Using Burp Suite as an automated scanner? Wondering right, even some pentesters do not prefer it, due to the fewer issues or the vulnerabilities it carries within. But what, if the burp scanner itself could identify the least common vulnerabilities along with core findings.

one of the most popular burp plugins **“Active Scan++”** thereby merges up with the burp’s scanner engine in order to enhance its scanning capabilities to identify the additional issues within an application.

Exploring & Initializing Active Scan++

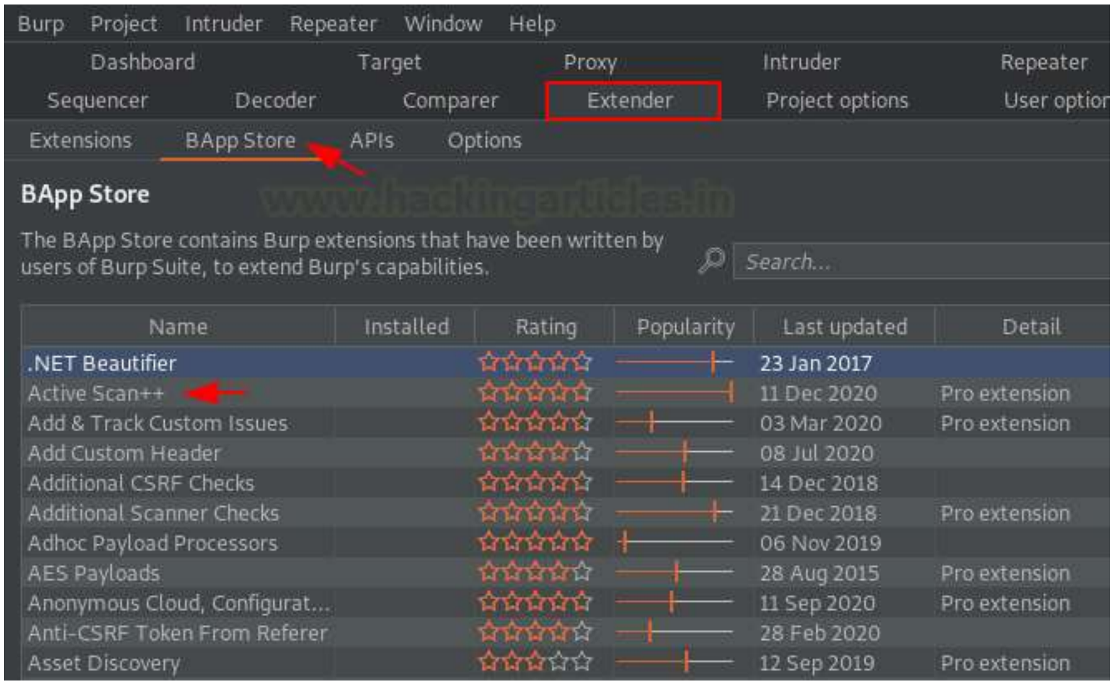
Advanced vulnerabilities require advanced scanning techniques.

Thereby, Active Scan++ is one of the most of most popular burp extensions designed for the **Burp’s Professional users** by **“James Kettle”** in order to improvise the burp’s active and passive scanning capabilities.

However, this plugin gets integrated within the burp scanner such that it could help in the issue discovering part for the **Host Header Attacks, Password Reset Poisoning, Cache Poisoning, DNS Rebinding, XML Injection, Arbitrary Header Injection, Template Injection, Blind Code Injection**, and the list goes on.

Moreover, this plugin also identifies the insertion points for **HTTP Basic Authentication.**

Being so much effective, so let’s find it out at the **bApp** store first. And we know where to navigate for that. Over at the **Extender** section, switch to the **bApp store** and then you’ll find this tool at the top with the highest rating.



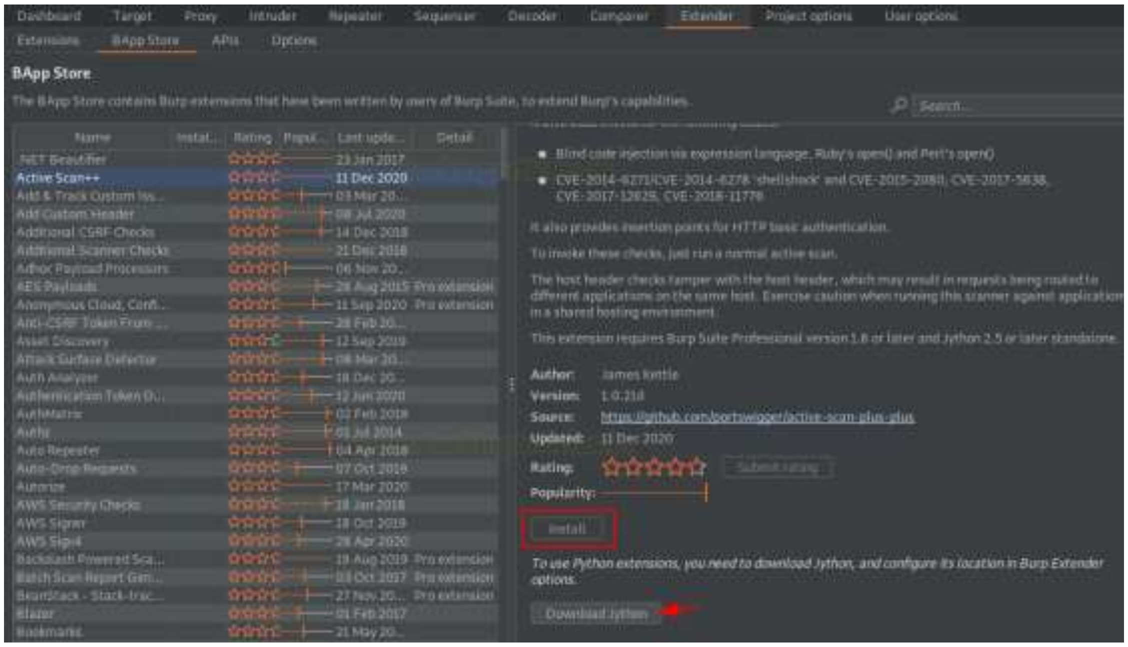
Burp Project Intruder Repeater Window Help
Dashboard Target Proxy Intruder Repeater
Sequencer Decoder Comparer Extender Project options User option
Extensions BApp Store APIs Options
www.hackingarticles.in
Search...

BApp Store

The BApp Store contains Burp extensions that have been written by users of Burp Suite, to extend Burp's capabilities.

Name	Installed	Rating	Popularity	Last updated	Detail
.NET Beautifier				23 Jan 2017	
Active Scan++				11 Dec 2020	Pro extension
Add & Track Custom Issues				03 Mar 2020	Pro extension
Add Custom Header				08 Jul 2020	
Additional CSRF Checks				14 Dec 2018	
Additional Scanner Checks				21 Dec 2018	Pro extension
Adhoc Payload Processors				06 Nov 2019	
AES Payloads				28 Aug 2015	Pro extension
Anonymous Cloud, Configurat...				11 Sep 2020	Pro extension
Anti-CSRF Token From Referer				28 Feb 2020	
Asset Discovery				12 Sep 2019	Pro extension

Let’s now switch to the left panel in order to identify the **Install** button. But wait!! It requires Jython, so let’s install and configure that first.



BApp Store

The BApp Store contains Burp extensions that have been written by users of Burp Suite, to extend Burp's capabilities.

Name	Instal...	Rating	Popul...	Last upde...	Detail
.NET Beautifier				23 Jan 2017	
Active Scan++				11 Dec 2020	
Add & Track Custom Iss...				03 Mar 20...	
Add Custom Header				08 Jul 2020	
Additional CSRF Checks				14 Dec 2018	
Additional Scanner Checks				21 Dec 2018	
Adhoc Payload Processors				06 Nov 20...	
AES Payloads				28 Aug 2015	Pro extension
Anonymous Cloud, Confi...				11 Sep 2020	Pro extension
Anti-CSRF Token From ...				28 Feb 20...	
Asset Discovery				12 Sep 2019	
Attack Surface Detector				08 Mar 20...	
Auth Analyzer				18 Dec 20...	
Authentication Token O...				12 Jun 2020	
AuthMatrix				02 Feb 2018	
Authz				01 Jul 2014	
Auto Repeater				04 Apr 2018	
Auto-Drop Requests				07 Oct 2019	
Autorize				17 Mar 2020	
AWS Security Checks				18 Jan 2018	
AWS Signer				18 Oct 2019	
AWS Sigv4				28 Apr 2020	
Backslash Powered Sca...				19 Aug 2019	Pro extension
Batch Scan Report Gen...				03 Oct 2017	Pro extension
BeanStack - Stack-trac...				27 Nov 20...	Pro extension
Blazer				01 Feb 2017	
Bookmarks				21 May 20...	

- Blind code injection via expression language, Ruby's open() and Perl's open()
- CVE-2014-6271/CVE-2014-6278 'shellshock' and CVE-2015-2080, CVE-2017-5638, CVE-2017-12629, CVE-2018-11776

It also provides insertion points for HTTP basic authentication.

To invoke these checks, just run a normal active scan.

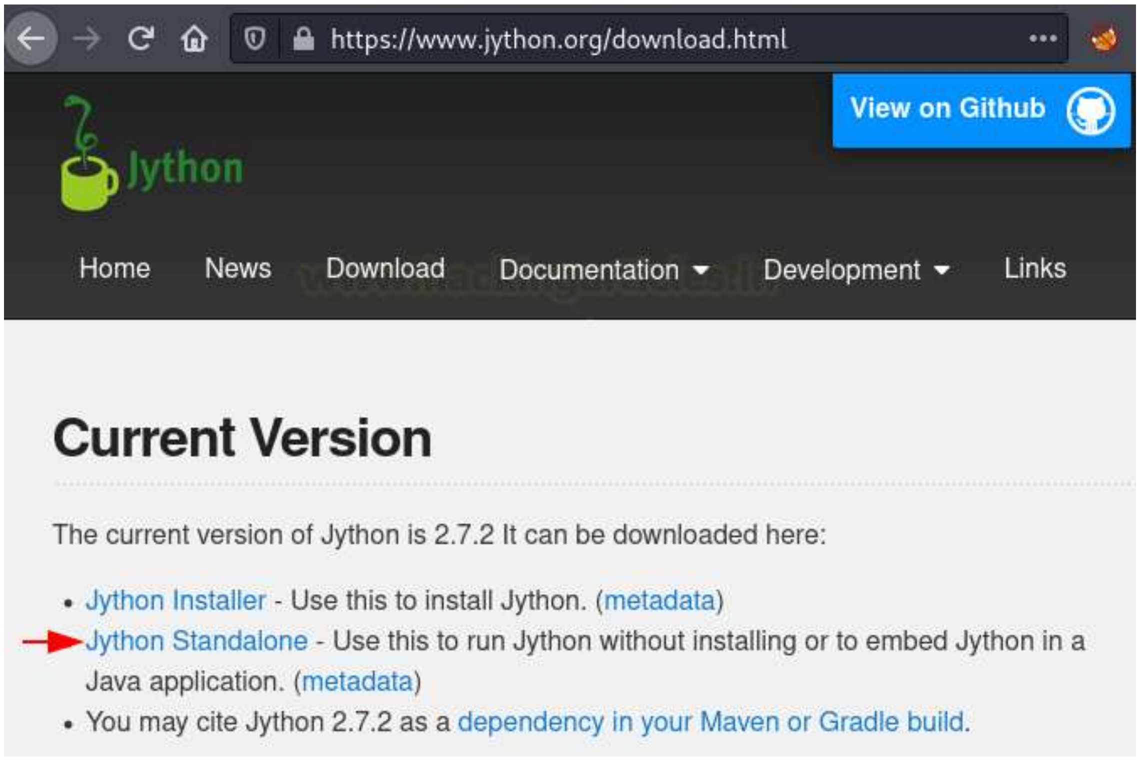
The host header checks tamper with the host header, which may result in requests being routed to different applications on the same host. Exercise caution when running this scanner against applications in a shared hosting environment.

This extension requires Burp Suite Professional version 1.6 or later and Jython 2.5 or later standalone.

Author: James Kettle
Version: 1.0.21d
Source: https://github.com/portswigger/active-scan-plus-plus
Updated: 11 Dec 2020
Rating:
Popularity:

To use Python extensions, you need to download Jython, and configure its location in Burp Extender options.

Head to Jython's official website, download the **Standalone Jython's** file.

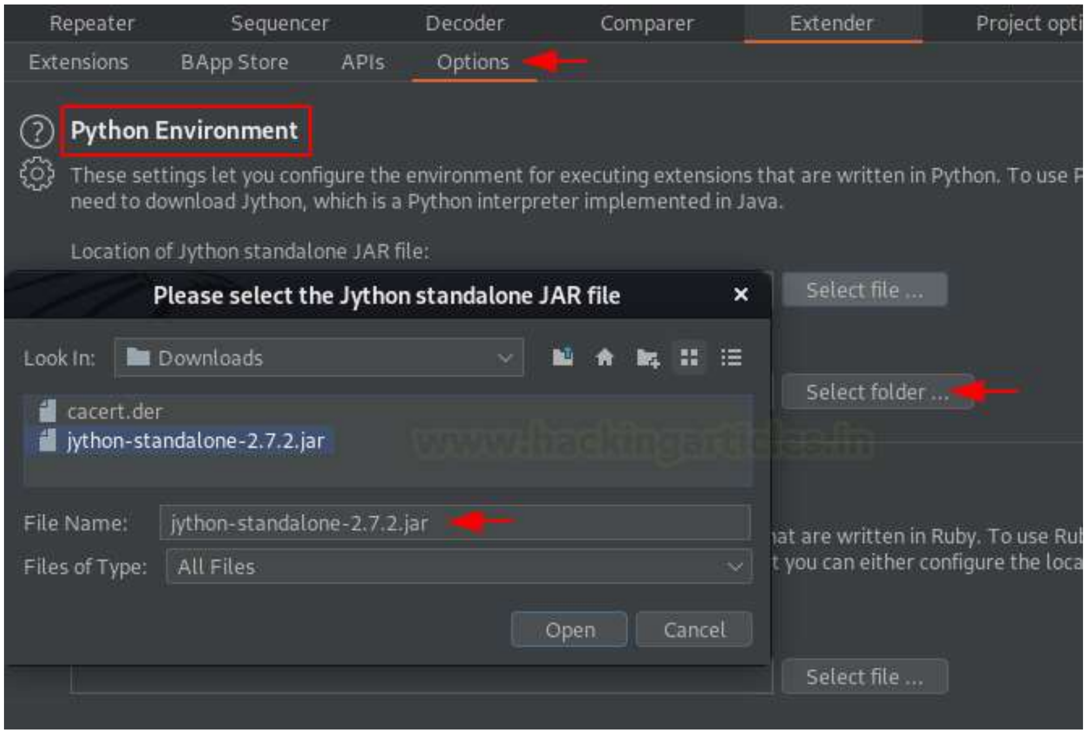


Current Version

The current version of Jython is 2.7.2 It can be downloaded here:

- Jython Installer - Use this to install Jython. (metadata)
- Jython Standalone - Use this to run Jython without installing or to embed Jython in a Java application. (metadata)
- You may cite Jython 2.7.2 as a dependency in your Maven or Gradle build.

As soon as the file got stored up over at the local machine, we'll embed it with our Burp Suite application. Back at the extender tab, navigate to the **Options** section there and scroll down for the **"Python Environment",** hit the **select file** button, and then opt for the downloaded file.



These settings let you configure the environment for executing extensions that are written in Python. To use P
need to download Jython, which is a Python interpreter implemented in Java.

Location of Jython standalone JAR file:

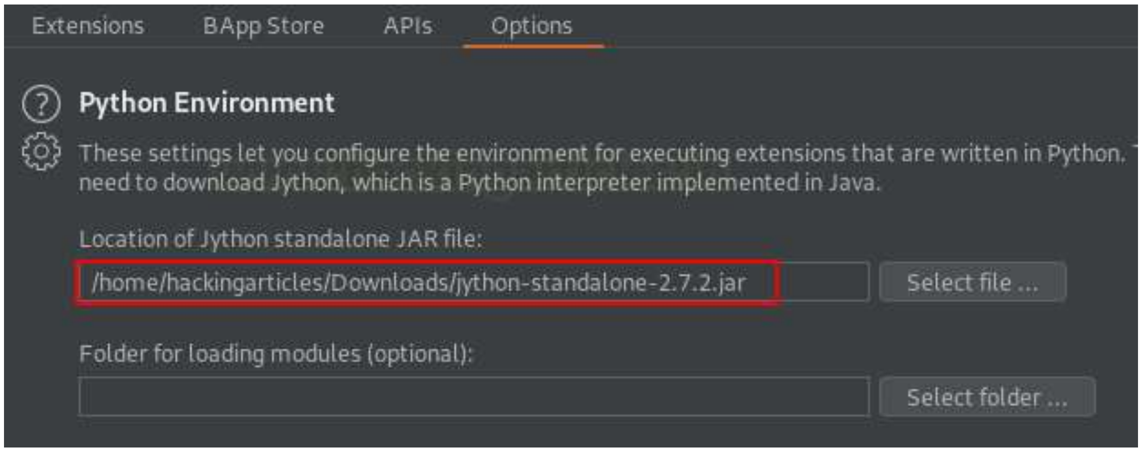
Please select the Jython standalone JAR file

Look In: Downloads

cacert.der
jython-standalone-2.7.2.jar

File Name: jython-standalone-2.7.2.jar
Files of Type: All Files

Once done with all this, you'll have your screen somewhat similar to the below image. But, the Jython's configuration is yet not over, **restart your burp** in order to get the changes reflected.

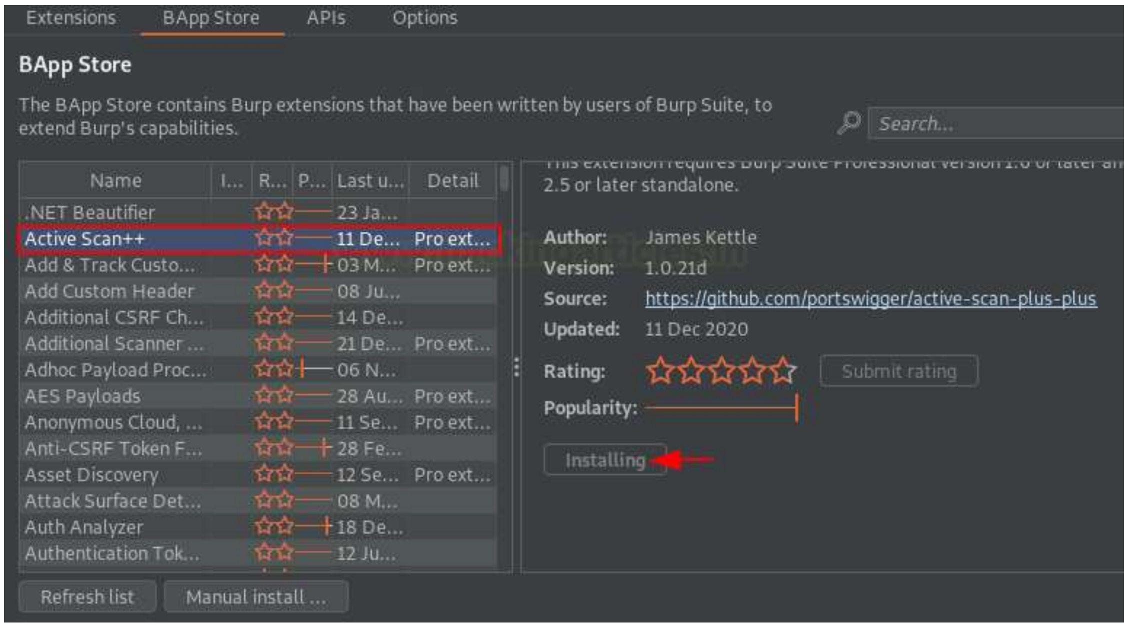


These settings let you configure the environment for executing extensions that are written in Python.
need to download Jython, which is a Python interpreter implemented in Java.

Location of Jython standalone JAR file:
/home/hackingarticles/Downloads/jython-standalone-2.7.2.jar

Folder for loading modules (optional):

Now head back to the bApp store and open the **Active Scan++** righ-side portal. From the below image you can see that the **Install** button is now active, let's fire it to initiate the installation.



BApp Store

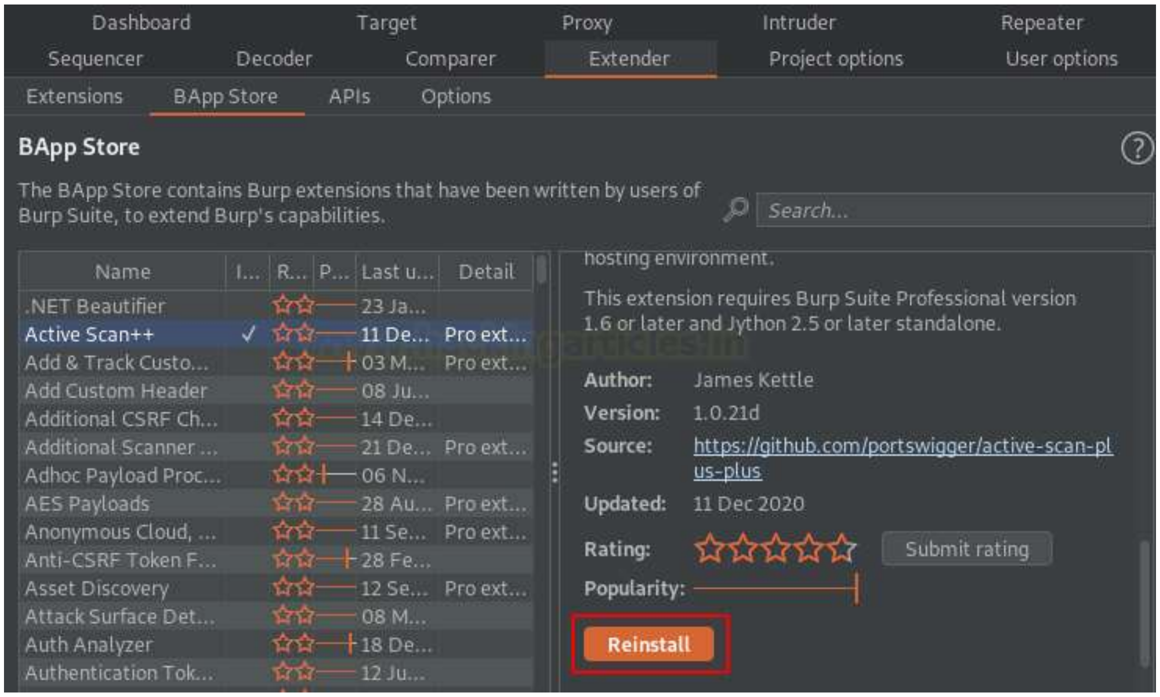
The BApp Store contains Burp extensions that have been written by users of Burp Suite, to extend Burp's capabilities.

Name	I...	R...	P...	Last u...	Detail
.NET Beautifier		☆☆		23 Ja...	
Active Scan++		☆☆		11 De...	Pro ext...
Add & Track Custo...		☆☆		03 M...	Pro ext...
Add Custom Header		☆☆		08 Ju...	
Additional CSRF Ch...		☆☆		14 De...	
Additional Scanner ...		☆☆		21 De...	Pro ext...
Adhoc Payload Proc...		☆☆☆		06 N...	
AES Payloads		☆☆		28 Au...	Pro ext...
Anonymous Cloud, ...		☆☆		11 Se...	Pro ext...
Anti-CSRF Token F...		☆☆		28 Fe...	
Asset Discovery		☆☆		12 Se...	Pro ext...
Attack Surface Det...		☆☆		08 M...	
Auth Analyzer		☆☆		18 De...	
Authentication Tok...		☆☆		12 Ju...	

2.5 or later standalone.

Author: James Kettle
Version: 1.0.21d
Source: https://github.com/portswigger/active-scan-plus-plus
Updated: 11 Dec 2020
Rating: ☆☆☆☆☆
Popularity:

Great!! We got the **Reinstall** button, seems like the extension had been set up successfully.



BApp Store

The BApp Store contains Burp extensions that have been written by users of Burp Suite, to extend Burp's capabilities.

Name	I...	R...	P...	Last u...	Detail
.NET Beautifier		☆☆		23 Ja...	
Active Scan++	✓	☆☆		11 De...	Pro ext...
Add & Track Custo...		☆☆		03 M...	Pro ext...
Add Custom Header		☆☆		08 Ju...	
Additional CSRF Ch...		☆☆		14 De...	
Additional Scanner ...		☆☆		21 De...	Pro ext...
Adhoc Payload Proc...		☆☆☆		06 N...	
AES Payloads		☆☆		28 Au...	Pro ext...
Anonymous Cloud, ...		☆☆		11 Se...	Pro ext...
Anti-CSRF Token F...		☆☆		28 Fe...	
Asset Discovery		☆☆		12 Se...	Pro ext...
Attack Surface Det...		☆☆		08 M...	
Auth Analyzer		☆☆		18 De...	
Authentication Tok...		☆☆		12 Ju...	

hosting environment.

This extension requires Burp Suite Professional version 1.6 or later and Jython 2.5 or later standalone.

Author: James Kettle
Version: 1.0.21d
Source: https://github.com/portswigger/active-scan-plus-plus
Updated: 11 Dec 2020
Rating: ☆☆☆☆☆
Popularity:

Enhancing the Audit Functionalities

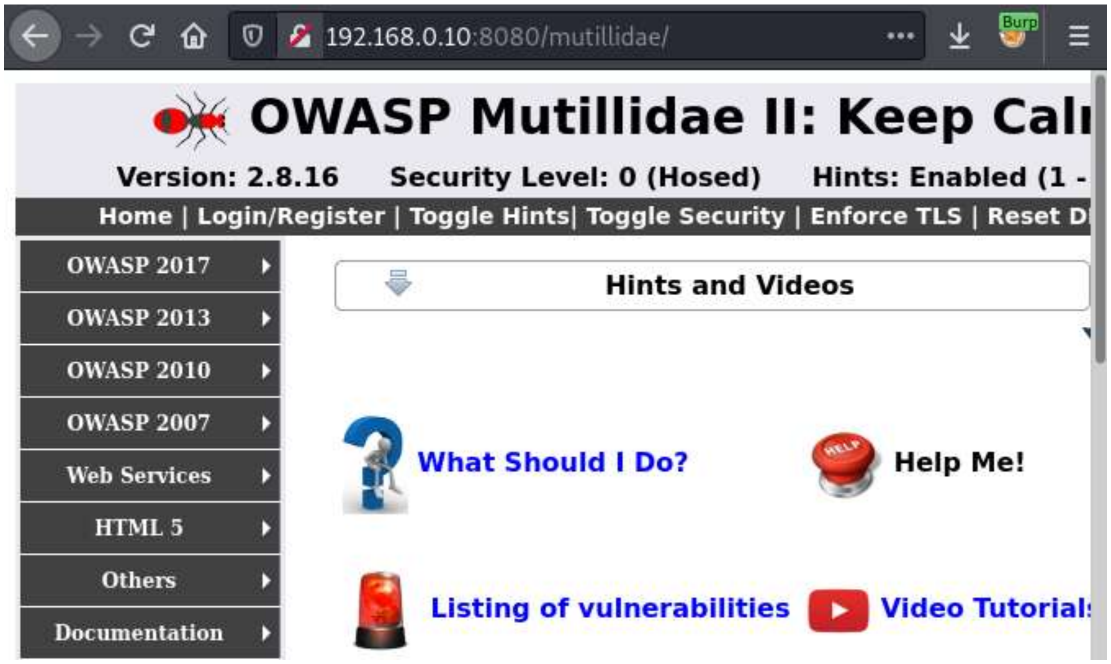
You might be wondering, like being the most popular, Active Scan++ should have its own place at the top panel, so where it is?

As discussed earlier that **Active Scan++ integrates with the burp's scanner** in order to assist it to identify additional vulnerabilities. Thereby, we do not have any specific location to find it. However, we can analyze its working while performing an active or passive scan.

So, let's see what additional it dumps out when we initiate a scan over at the entire application.

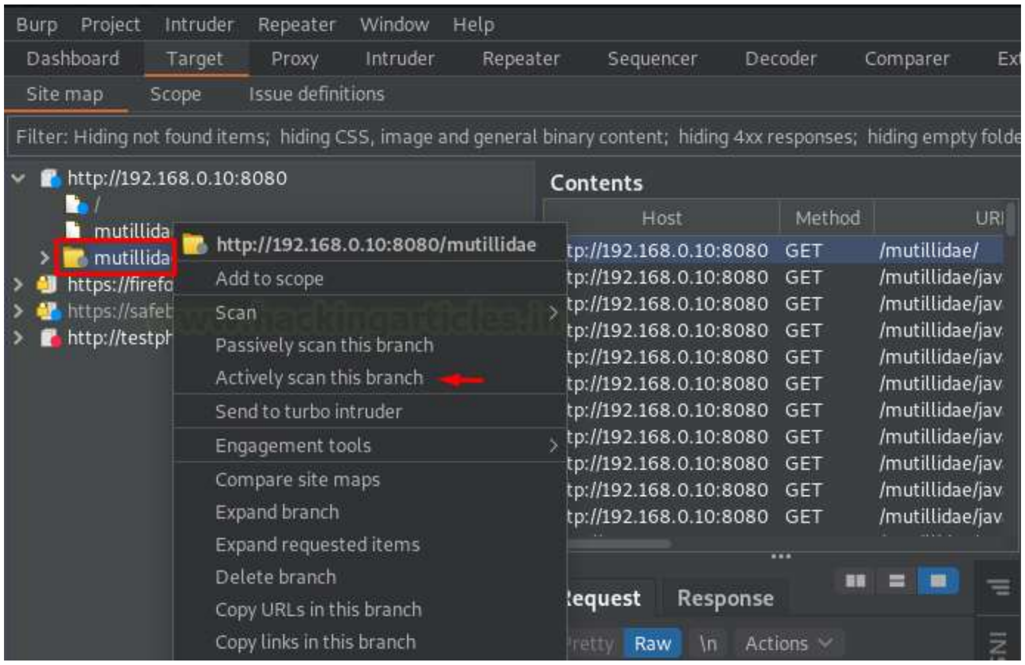
Audit the application

Turn **On** the Browser's **Proxy Service** and then surf the OWASP's Mutillidae vulnerable application.


192.168.0.10:8080/mutillidae/
OWASP Mutillidae II: Keep Cal
Version: 2.8.16 Security Level: 0 (Hosed) Hints: Enabled (1 -
Home | Login/Register | Toggle Hints| Toggle Security | Enforce TLS | Reset D
OWASP 2017
OWASP 2013
OWASP 2010
OWASP 2007
Web Services
HTML 5
Others
Documentation
Hints and Videos
What Should I Do?
Help Me!
Listing of vulnerabilities
Video Tutorial

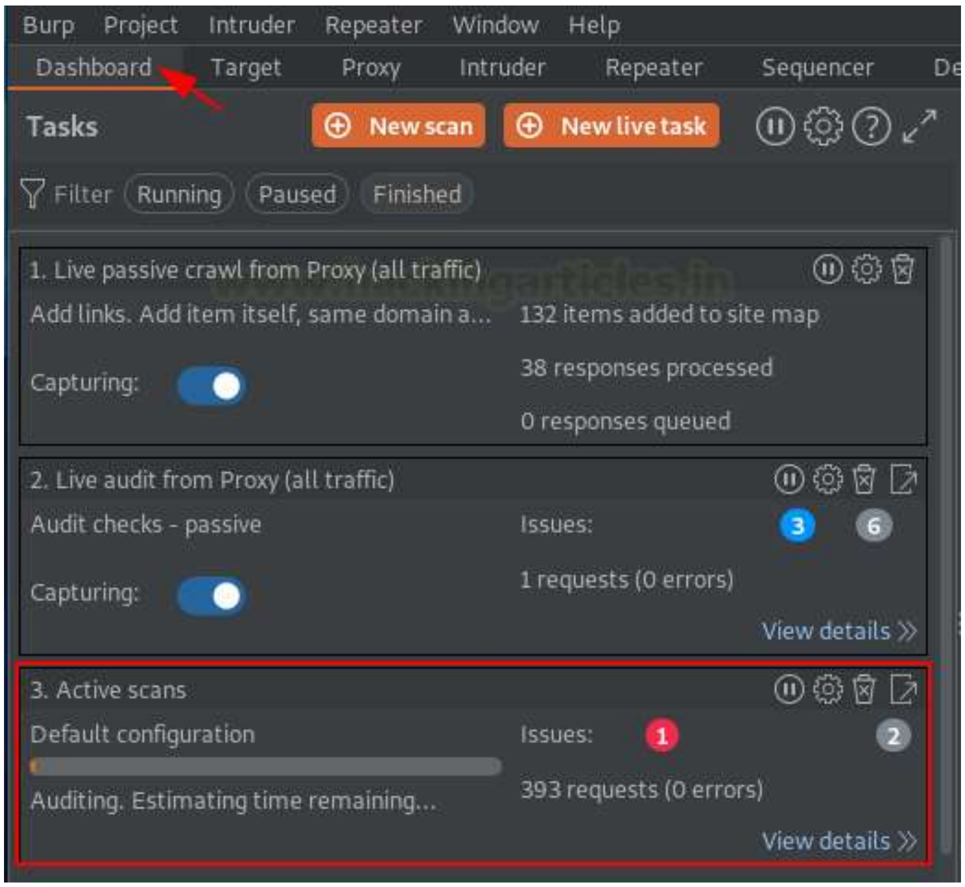
Now, let's navigate to the **Target** option over at our burp suite monitor, further head to the **Site map**.

The Left panel carries up the web application's hierarchy, let's opt for the **root branch**, and then we'll hit a right-click to opt **"Actively scan this branch"** in order to share the application for the scanning part.


Burp Project Intruder Repeater Window Help
Dashboard Target Proxy Intruder Repeater Sequencer Decoder Comparer Ex
Site map Scope Issue definitions
Filter: Hiding not found items; hiding CSS, image and general binary content; hiding 4xx responses; hiding empty folde
http://192.168.0.10:8080
/
mutillida
mutillida
https://firefo
https://safeb
http://testph
http://192.168.0.10:8080/mutillidae
Add to scope
Scan
Passively scan this branch
Actively scan this branch
Send to turbo intruder
Engagement tools
Compare site maps
Expand branch
Expand requested items
Delete branch
Copy URLs in this branch
Copy links in this branch
Contents
Host
Method
URL
Request
Response
Pretty
Raw
\n
Actions

Host	Method	URL
tp://192.168.0.10:8080	GET	/mutillidae/
tp://192.168.0.10:8080	GET	/mutillidae/jav
tp://192.168.0.10:8080	GET	/mutillidae/jav
tp://192.168.0.10:8080	GET	/mutillidae/jav
tp://192.168.0.10:8080	GET	/mutillidae/jav
tp://192.168.0.10:8080	GET	/mutillidae/jav
tp://192.168.0.10:8080	GET	/mutillidae/jav
tp://192.168.0.10:8080	GET	/mutillidae/jav
tp://192.168.0.10:8080	GET	/mutillidae/jav
tp://192.168.0.10:8080	GET	/mutillidae/jav
tp://192.168.0.10:8080	GET	/mutillidae/jav

As soon as do so, we got our auditing task aligned at the **"Dashboard".**


Burp Project Intruder Repeater Window Help
Dashboard Target Proxy Intruder Repeater Sequencer De
Tasks
New scan
New live task
Filter Running Paused Finished

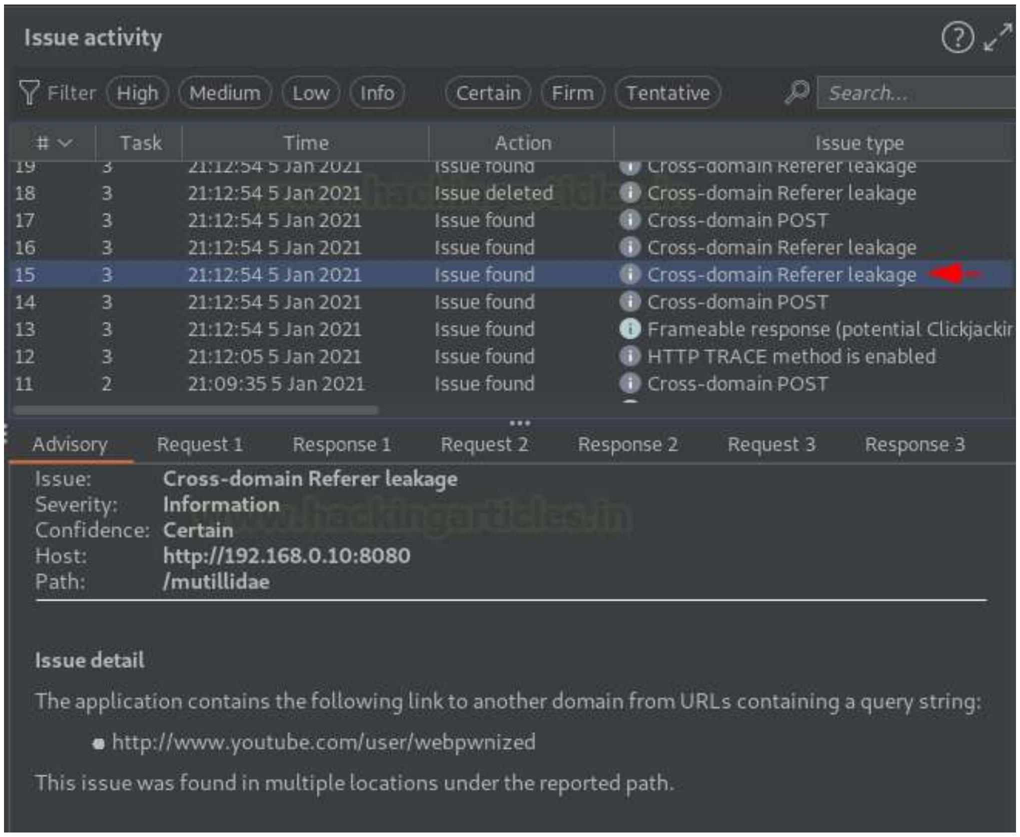
1. Live passive crawl from Proxy (all traffic)
Add links. Add item itself, same domain a...
132 items added to site map
Capturing:
38 responses processed
0 responses queued

2. Live audit from Proxy (all traffic)
Audit checks - passive
Issues: 3 6
Capturing:
1 requests (0 errors)
View details »

3. Active scans
Default configuration
Issues: 1 2
Auditing. Estimating time remaining...
393 requests (0 errors)
View details »

And within a few minutes, you can see a number of issues segregated at the right panel, let's check them out.

You might find most of the issues discovered with the burp's scanner but there are some like cache poisoning, DNS rebinding, Host header Injection, all these additional ones are identified with the Active Scan++.



Issue activity

Filter High Medium Low Info Certain Firm Tentative Search...

#	Task	Time	Action	Issue type
19	3	21:12:54 5 Jan 2021	Issue found	Cross-domain Referer leakage
18	3	21:12:54 5 Jan 2021	Issue deleted	Cross-domain Referer leakage
17	3	21:12:54 5 Jan 2021	Issue found	Cross-domain POST
16	3	21:12:54 5 Jan 2021	Issue found	Cross-domain Referer leakage
15	3	21:12:54 5 Jan 2021	Issue found	Cross-domain Referer leakage
14	3	21:12:54 5 Jan 2021	Issue found	Cross-domain POST
13	3	21:12:54 5 Jan 2021	Issue found	Frameable response (potential Clickjackir
12	3	21:12:05 5 Jan 2021	Issue found	HTTP TRACE method is enabled
11	2	21:09:35 5 Jan 2021	Issue found	Cross-domain POST

Advisory Request 1 Response 1 Request 2 Response 2 Request 3 Response 3

Issue:	Cross-domain Referer leakage
Severity:	Information
Confidence:	Certain
Host:	http://192.168.0.10:8080
Path:	/mutillidae

Issue detail

The application contains the following link to another domain from URLs containing a query string:

- http://www.youtube.com/user/webpwnized

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

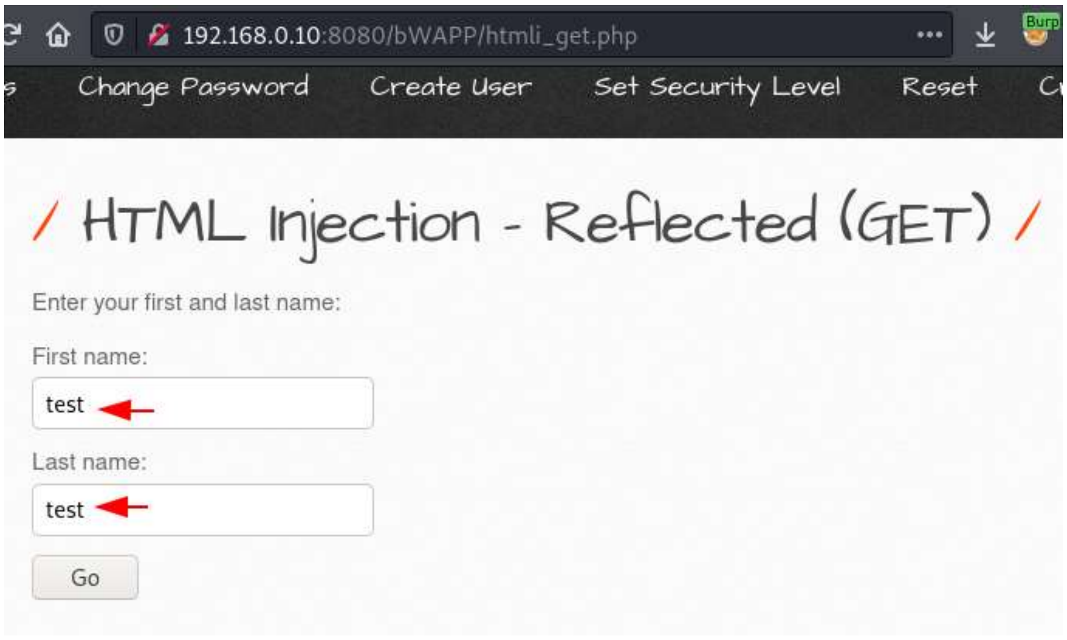
Auditing specific Injection Points

What if, you don't want to test the entire application's branch or a specific web page, but you want an injection point to be audited and if it's possible then does ***Active Scan++*** still collaborate with the burp scanner?

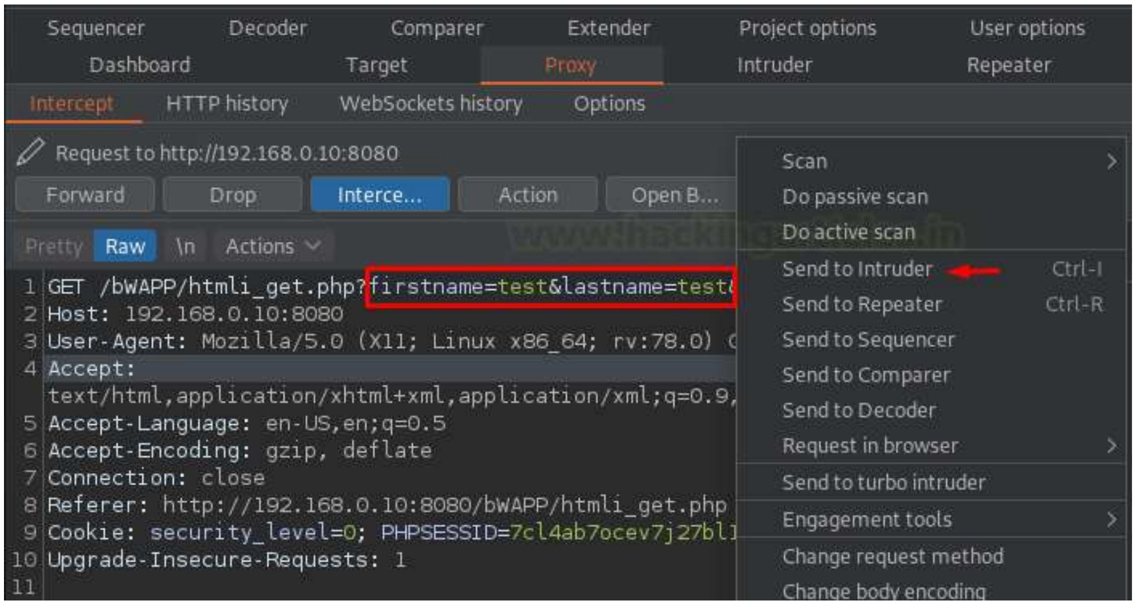
As soon as we hit the install button at the bApp store, the very first-second Active Scan++ got bound with the Burp's Scanner. So, whether it's about auditing the entire application or a specific injection point, the Active Scan++ is always involved within it.

And about the Injection point auditing, so let's do that.

Initiate the bWAPP application with **bee: bug** and then navigate to the **HTML Injection (Reflected)** webpage. Further, we'll enter some test credentials and hit the **Go** button with our **Proxy Service** turned **"ON"**

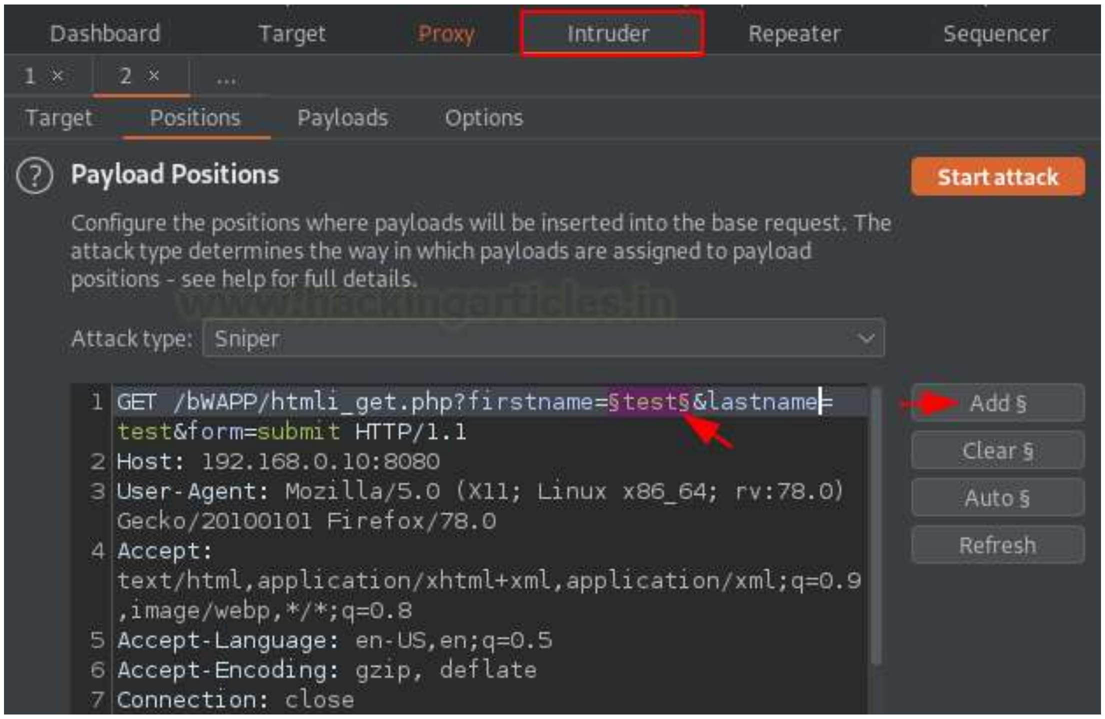

192.168.0.10:8080/bWAPP/htmli_get.php
Change Password
Create User
Set Security Level
Reset
/ HTML Injection - Reflected (GET) /
Enter your first and last name:
First name:
test
Last name:
test
Go

Over at our burp suite monitor, we got the ongoing HTTP request captured, let's share it with the intruder.



```
GET /bWAPP/htmli_get.php?firstname=test&lastname=test
Host: 192.168.0.10:8080
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:78.0) 
Accept:
text/html,application/xhtml+xml,application/xml;q=0.9,
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Connection: close
Referer: http://192.168.0.10:8080/bWAPP/htmli_get.php
Cookie: security_level=0; PHPSESSID=7cl4ab7ocev7j27bl1
Upgrade-Insecure-Requests: 1
```

Once done, let's navigate to the **Position section** over at the intruder tab in order to set the injection point. Simply clear the default selections and add the one you want the scanner to audit.



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.

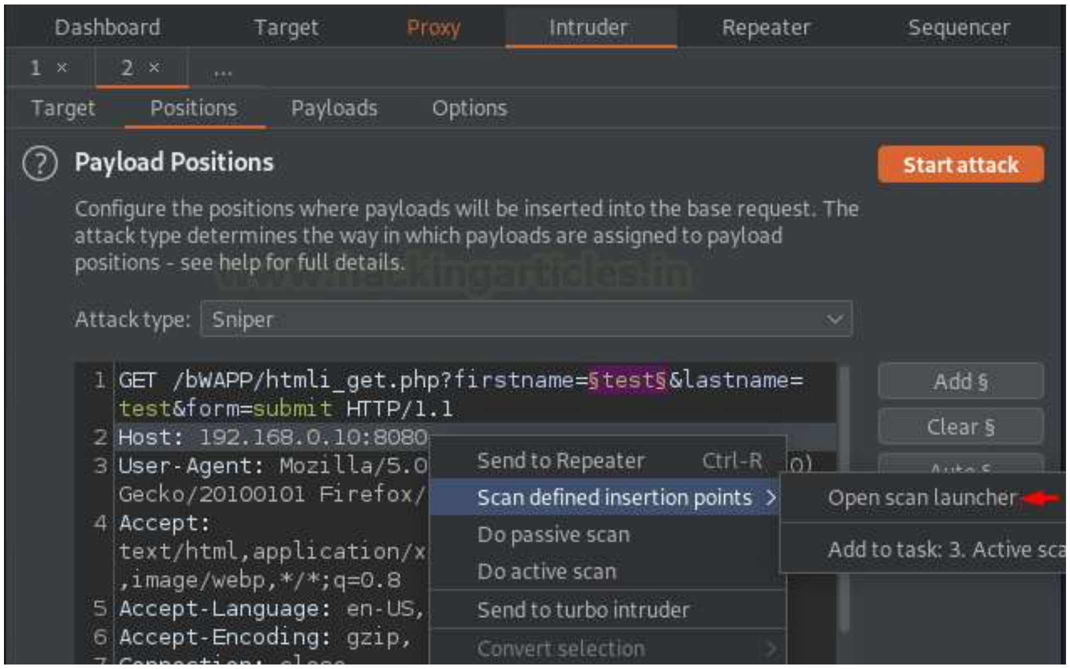
Attack type: Sniper

```
GET /bWAPP/htmli_get.php?firstname=§test§&lastname=
test&form=submit HTTP/1.1
Host: 192.168.0.10:8080
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:78.0)
Gecko/20100101 Firefox/78.0
Accept:
text/html,application/xhtml+xml,application/xml;q=0.9
,image/webp,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Connection: close
```

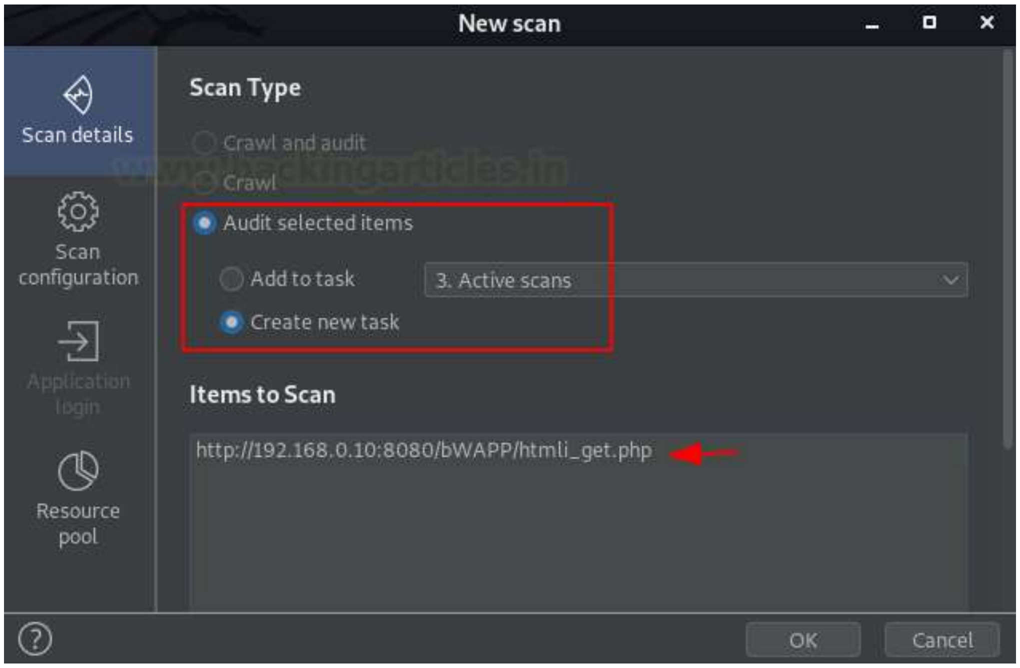
What now? A simple right-click is always our helping hand. So, let’s hit the mouse right button and then opt **“Scan defined insertion points”.**

As we haven’t deleted our previous task, thereby the burp drops two option either to send this request with the ongoing Audit or to start a new one for this.

Let’s hit the **Open Scan launcher** in order to start and customize the new scan.



As soon as we do so, we got a window that popped up in front of us stating **“New Scan”**. Let’s move forward with the default configuration by hitting the **OK** button.



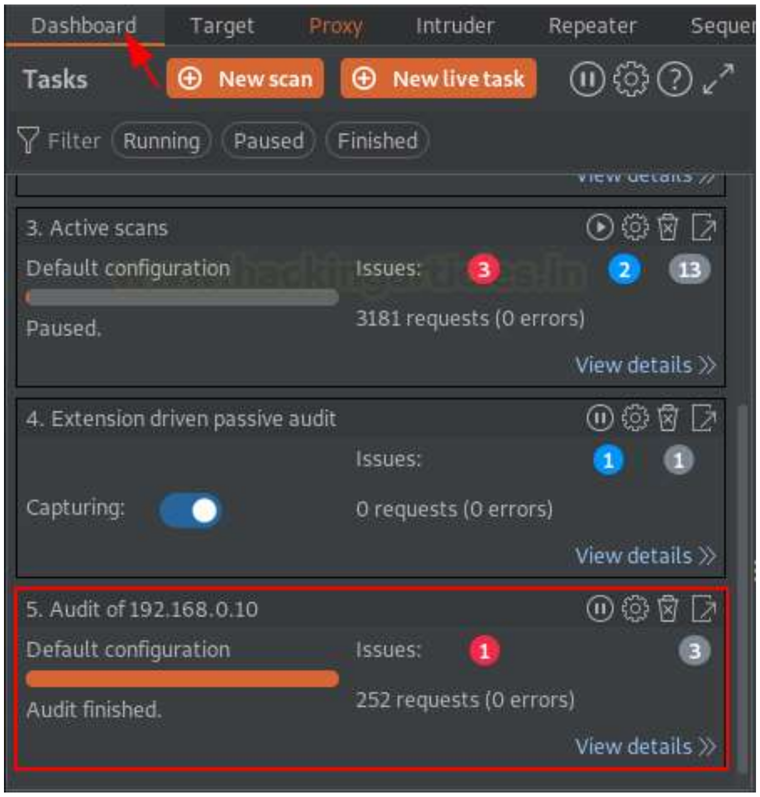
Scan Type

Crawl and audit
Crawl
[✓ Audit selected items]
Add to task: 3. Active scans
[✓ Create new task]

Items to Scan

http://192.168.0.10:8080/bWAPP/htmli_get.php

From the below image we can see that our scanner captured about **3 Basic** and **1 Critical issue** by sharing about 250+ requests at the application's injection point.

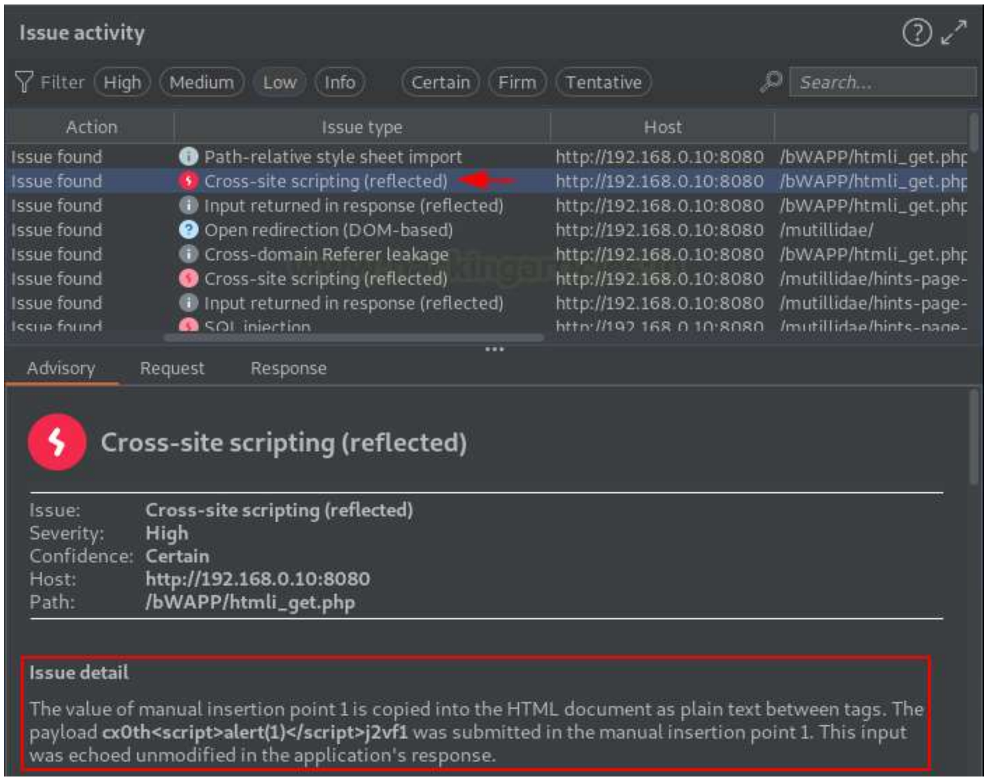


3. Active scans
Default configuration — Issues: 3, 2, 13
Paused.
3181 requests (0 errors)

4. Extension driven passive audit
Issues: 1, 1
Capturing:
0 requests (0 errors)

5. Audit of 192.168.0.10
Default configuration — Issues: 1, 3
Audit finished.
252 requests (0 errors)

Time to analyze. From the below image, we can see that the burp scanner tested Cross-Site Scripting for our injection point and dumped the issue details with the exploiting payload and the mitigation steps.



Issue activity

Filter High Medium Low Info Certain Firm Tentative Search...

Action	Issue type	Host	
Issue found	Path-relative style sheet import	http://192.168.0.10:8080	/bWAPP/htmli_get.php
Issue found	Cross-site scripting (reflected)	http://192.168.0.10:8080	/bWAPP/htmli_get.php
Issue found	Input returned in response (reflected)	http://192.168.0.10:8080	/bWAPP/htmli_get.php
Issue found	Open redirection (DOM-based)	http://192.168.0.10:8080	/mutillidae/
Issue found	Cross-domain Referer leakage	http://192.168.0.10:8080	/bWAPP/htmli_get.php
Issue found	Cross-site scripting (reflected)	http://192.168.0.10:8080	/mutillidae/hints-page-
Issue found	Input returned in response (reflected)	http://192.168.0.10:8080	/mutillidae/hints-page-
Issue found	SQL injection	http://192.168.0.10:8080	/mutillidae/hints-page-

Advisory Request Response

Cross-site scripting (reflected)

Issue:	**Cross-site scripting (reflected)**
Severity:	**High**
Confidence:	**Certain**
Host:	**http://192.168.0.10:8080**
Path:	**/bWAPP/htmli_get.php**

Issue detail

The value of manual insertion point 1 is copied into the HTML document as plain text between tags. The payload **cx0th<script>alert(1)</script>j2vf1** was submitted in the manual insertion point 1. This input was echoed unmodified in the application's response.