1. ZAP:

Client-side bypassing:

Total time/vulnerabilities:

Time to plan and run test cases: 7 hours 51 minutes

Vulnerabilities found: 2

Number of vulnerabilities/hour : 0.255 vulns/hours

1. **ASVS V5.1 Input Validation**

Unique ID: 5.1.1-0

CWE 235: Improper Handling of Extra Parameters

Repeatable steps:

1. If not already running, start/run OpenEMR.

2. Open ZAP and create a new connection by selecting “Manual Explore”.

3. In “URL to explore”, enter: ‘http://localhost’. Leave “Enable HUD” unchecked, and then press “Launch Browser” to open your browser of choice. When the browser opens, if “Welcome to the ZAP HUD” appears, select “Continue to your target”.

4. Login with your admin account, unless changed, the credentials should be “admin” for the username, and “pass” for the password.

5. Select Finder in the topmost bar.

6. In the text box after “Search:” on the right-hand side, input “test”.

7. Back in ZAP, in the left menu, navigate to the recent request:

Under “Sites”, click “http://localhost”.

Next, in the dropdown, select “interface”.

Next, in the dropdown, select “main”.

Next, in the dropdown, select “finder”.

Now, right click on “GET:dynamic_finder_ajax.php …” and select “Break…”

8. In the Add Breakpoint dialog, change the value to http://localhost/interface/main/finder/dynamic_finder_ajax.php

9. Select Save

10. Go back to OpenEMR and input the letter t into the same Search bar.

11. In ZAP, go to the Header portion of the Break text box. Find the string &sSearch=t&, replace it with &sSearch=' or 1=1 --&

12. Then, select the Play button on ZAP until the button is disabled.

Expected results:

1. Deny the request or show no patients on the table.

2. ASVS V5.1 Input Validation

Unique ID: 5.1.1-1

CWE 235: Improper Handling of Extra Parameters

Repeatable steps:

1. If not already running, start/run OpenEMR.

2. Open ZAP and create a new connection by selecting “Manual Explore”.

3. In “URL to explore”, enter: ‘http://localhost’. Leave “Enable HUD” unchecked, and then press “Launch Browser” to open your browser of choice. When the browser opens, if “Welcome to the ZAP HUD” appears, select “Continue to your target”.

4. Login with your admin account, unless changed, the credentials should be “admin” for the username, and “pass” for the password.

5. Select Reports, Then Visits. Then, Daily Report.

6. Select Submit.

7. Back in ZAP, in the left menu, navigate to the recent request:

Under “Sites”, click “http://localhost”.

Next, in the dropdown, select “interface”.

Next, in the dropdown, select “reports”.

Now, right click on “GET:daily_summary_report.php …” and select “Break…”

8. Select Save in the Add Breakpoint dialog.

9. Go back to OpenEMR and select the Submit button again..

10. In ZAP, go to the Body portion of the Break text boxes. Change &form_provider=& to &form_provider=' or 1=1 --& .

11. Then, select the Play button on ZAP until the button is disabled.

Expected results:

1. Deny the request and/or show nothing in the table.

3. ASVS V5.3 Output Encoding and Injection Prevention

Unique ID: 5.3.3-0	CWE 79: Product does not neutralize input.

Repeatable steps:

1. If not already running, start/run OpenEMR.

2. Open ZAP and create a new connection by selecting “Manual Explore”.

3. In “URL to explore”, enter: ‘http://localhost’. Leave “Enable HUD” unchecked, and then press “Launch Browser” to open your browser of choice. When the browser opens, if “Welcome to the ZAP HUD” appears, select “Continue to your target”.

4. Login with your admin account, unless changed, the credentials should be “admin” for the username, and “pass” for the password.

5. At the top right of the page, select the search bar, enter “test” and press the search icon.

6. Back in ZAP, in the left menu, navigate to the recent request:

Under “Sites”, click “http://localhost”.

Next, in the dropdown, select “interface”.

Next, in the dropdown, select “main”.

Next, in the dropdown, select “finder”.

Now, right click on “GET:dynamic_finder.php(search_any)” and select “Break…”

7. In the “Add Breakpoint” box,edit the “String” like so:

http://localhost/interface/main/finder/dynamic_finder.php\?search_any

Then press “Save”

8. Go back to your web browser where you made the search request. Enter in the following into your Search Bar:

```
<script>alert('XSS')</script>
```

Then press the search icon.

9. ZAP should bring up a “Break” tab located in the menu where the “Quick Start” tab is. On this screen, you should see a menu that says “Method” with a drop down icon. Click it and select “POST”

10. You should now see: “search_any=<script>alert('XSS')</script>”

11. Now, back in your browser, continue to press “Step” until the request is finished.

Expected results:

1. The script should not execute and no alert box should appear.

4. ASVS V5.3 Output Encoding and Injection Prevention

Unique ID: 5.3.8-0

CWE 78: Product does not Neutralize OS Command input.

Repeatable steps:

1. If not already running, start/run OpenEMR.

2. Open ZAP and create a new connection by selecting “Manual Explore”.

3. In “URL to explore”, enter: ‘http://localhost’. Leave “Enable HUD” unchecked, and then press “Launch Browser” to open your browser of choice. When the browser opens, if “Welcome to the ZAP HUD” appears, select “Continue to your target”.

4. Login with your admin account, unless changed, the credentials should be “admin” for the username, and “pass” for the password.

5. At the top right of the page, select the search bar, enter “test” and press the search icon.

6. Back in ZAP, in the left menu, navigate to the recent request:

Under “Sites”, click “http://localhost”.

Next, in the dropdown, select “interface”.

Next, in the dropdown, select “main”.

Next, in the dropdown, select “finder”.

Now, right click on “GET:dynamic_finder.php(search_any)” and select “Break…”

7. In the “Add Breakpoint” box,edit the “String” like so:

http://localhost/interface/main/finder/dynamic_finder.php\?search_any

Then press “Save”

8. Go back to your web browser where you made the search request. Enter in the following into your Search Bar:
; ls or | dir

Then press the search icon.

9. ZAP should bring up a “Break” tab located in the menu where the “Quick Start” tab is. On this screen, you should see a menu that says “Method” with a drop down icon. Click it and select “POST”

10. You should now see: “search_any=; ls or | dir”

11. Now, back in your browser, continue to press “Step” until the request is finished.

Expected results:

1. The OS command should not execute and no data should appear from the server.

5. ASVS V5.1 Input Validation

Unique ID: 5.1.5-0

CWE 601: Link to external sites are accepted from the user.

Repeatable steps:

1. If not already running, start/run OpenEMR.

2. Open ZAP and create a new connection by selecting “Manual Explore”.

3. In “URL to explore”, enter: ‘http://localhost’. Leave “Enable HUD” unchecked, and then press “Launch Browser” to open your browser of choice. When the browser opens, if “Welcome to the ZAP HUD” appears, select “Continue to your target”.

4. Login with your admin account, unless changed, the credentials should be “admin” for the username, and “pass” for the password.

5. At the top right of the page, select the search bar, enter “test” and press the search icon.

6. Back in ZAP, in the left menu, navigate to the recent request:

Under “Sites”, click “http://localhost”.

Next, in the dropdown, select “interface”.

Next, in the dropdown, select “main”.

Next, in the dropdown, select “finder”.

Now, right click on “GET:dynamic_finder.php(search_any)” and select “Break…”

7. In the “Add Breakpoint” box,edit the “String” like so:

http://localhost/interface/main/finder/dynamic_finder.php\?search_any

Then press “Save”

8. Go back to your web browser where you made the search request. Enter in the following into your Search Bar:
 test

Then press the search icon.

9. ZAP should bring up a “Break” tab located in the menu where the “Quick Start” tab is. You should see the following:

GET
http://localhost/interface/main/finder/dynamic_finder.php?search_any=%3B%20ls%20or%20%7C%20dir HTTP/1.1

10. Change it to: GET https://www.google.com HTTP/1.1

11. In ZAP, at the top menu bar, find the button that looks like a play button, that when hovered over, says: “Submit and Continue to NextBreakpoint” and press it.

12. Now, back in your browser, continue to press “Step” until the request is finished.

Expected results:

1. The page should return a message saying “No matching records found”.

Fuzzing

Total time/vulnerabilities Fuzzing:

Time to run ZAP scan: 22 minutes

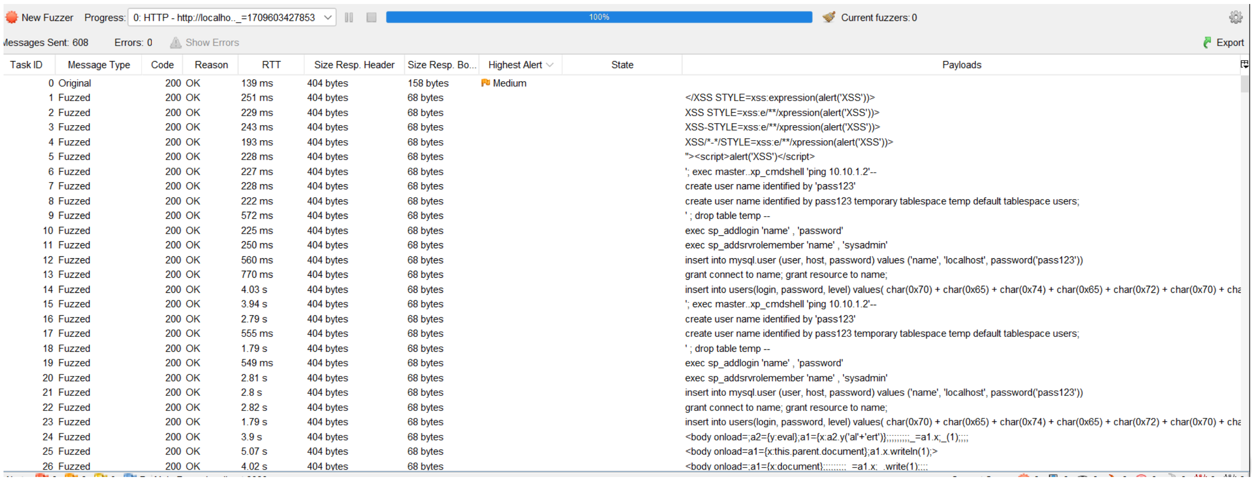
Time run test cases: 3 hours 57 minutes

Vulnerabilities found: 5

Number of vulnerabilities/hour : 1.266

Test 1:

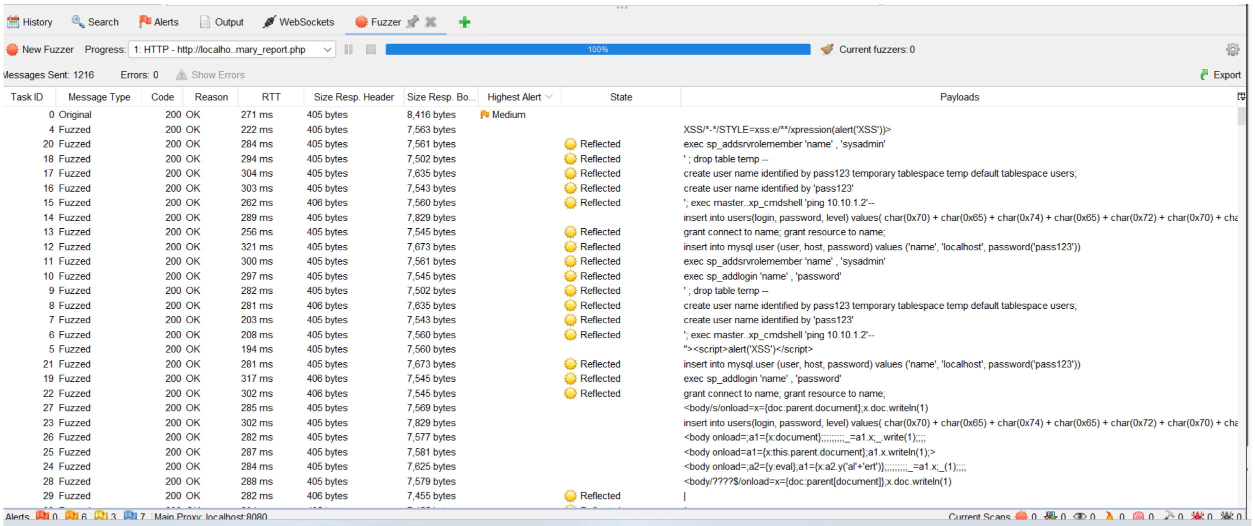
Screenshot of the fuzzing results for test case 1 for the 4 rulesets.



Task ID	Message Type	Code	Reason	RTT	Size Resp. Header	Size Resp. Bo...	Highest Alert	State	Payloads
0	Original	200	OK	139 ms	404 bytes	158 bytes	Medium		
1	Fuzzed	200	OK	251 ms	404 bytes	68 bytes			</XSS STYLE=xss:expression(alert('XSS'))>
2	Fuzzed	200	OK	229 ms	404 bytes	68 bytes			XSS STYLE=xss:e/**/xpression(alert('XSS'))>
3	Fuzzed	200	OK	243 ms	404 bytes	68 bytes			XSS-STYLE=xss:e/**/xpression(alert('XSS'))>
4	Fuzzed	200	OK	193 ms	404 bytes	68 bytes			XSS/*-*/STYLE=xss:e/**/xpression(alert('XSS'))>
5	Fuzzed	200	OK	228 ms	404 bytes	68 bytes			"><script>alert('XSS')</script>
6	Fuzzed	200	OK	227 ms	404 bytes	68 bytes			'; exec master..xp_cmdshell 'ping 10.10.1.2'--
7	Fuzzed	200	OK	228 ms	404 bytes	68 bytes			create user name identified by 'pass123'
8	Fuzzed	200	OK	222 ms	404 bytes	68 bytes			create user name identified by pass123 temporary tablespace temp default tablespace users;
9	Fuzzed	200	OK	572 ms	404 bytes	68 bytes			' ; drop table temp --
10	Fuzzed	200	OK	225 ms	404 bytes	68 bytes			exec sp_addlogin 'name' , 'password'
11	Fuzzed	200	OK	250 ms	404 bytes	68 bytes			exec sp_addsrvrolemember 'name' , 'sysadmin'
12	Fuzzed	200	OK	560 ms	404 bytes	68 bytes			insert into mysql.user (user, host, password) values ('name', 'localhost', password('pass123'))
13	Fuzzed	200	OK	770 ms	404 bytes	68 bytes			grant connect to name; grant resource to name;
14	Fuzzed	200	OK	4.03 s	404 bytes	68 bytes			insert into users(login, password, level) values(char(0x70) + char(0x65) + char(0x74) + char(0x65) + char(0x72) + char(0x70) + cha
15	Fuzzed	200	OK	3.94 s	404 bytes	68 bytes			'; exec master..xp_cmdshell 'ping 10.10.1.2'--
16	Fuzzed	200	OK	2.79 s	404 bytes	68 bytes			create user name identified by 'pass123'
17	Fuzzed	200	OK	555 ms	404 bytes	68 bytes			create user name identified by pass123 temporary tablespace temp default tablespace users;
18	Fuzzed	200	OK	1.79 s	404 bytes	68 bytes			' ; drop table temp --
19	Fuzzed	200	OK	549 ms	404 bytes	68 bytes			exec sp_addlogin 'name' , 'password'
20	Fuzzed	200	OK	2.81 s	404 bytes	68 bytes			exec sp_addsrvrolemember 'name' , 'sysadmin'
21	Fuzzed	200	OK	2.8 s	404 bytes	68 bytes			insert into mysql.user (user, host, password) values ('name', 'localhost', password('pass123'))
22	Fuzzed	200	OK	2.82 s	404 bytes	68 bytes			grant connect to name; grant resource to name;
23	Fuzzed	200	OK	1.79 s	404 bytes	68 bytes			insert into users(login, password, level) values(char(0x70) + char(0x65) + char(0x74) + char(0x65) + char(0x72) + char(0x70) + cha
24	Fuzzed	200	OK	3.9 s	404 bytes	68 bytes			<body onload=;a2={y:eval};a1={x:a2.y('al'+'ert')};;;;;;;;;_=a1.x;_(1);;;;
25	Fuzzed	200	OK	5.07 s	404 bytes	68 bytes			<body onload=a1={x:this.parent.document};a1.x.writeln(1);>
26	Fuzzed	200	OK	4.02 s	404 bytes	68 bytes			<body onload=;a1={x:document};;;;;;;;;_=a1.x;_.write(1);;;;

Test 2:

Screenshot



Progress: 1: HTTP - http://localho..mary_report.php 100% Current fuzzers: 0

Messages Sent: 1216 Errors: 0

Task ID	Message Type	Code	Reason	RTT	Size Resp. Header	Size Resp. Bo...	Highest Alert	State	Payloads
0	Original	200	OK	271 ms	405 bytes	8,416 bytes	Medium		
4	Fuzzed	200	OK	222 ms	405 bytes	7,563 bytes			XSS/*-*/STYLE=xss:e/**/xpression(alert('XSS'))>
20	Fuzzed	200	OK	284 ms	405 bytes	7,561 bytes		Reflected	exec sp_addsrvrolemember 'name' , 'sysadmin'
18	Fuzzed	200	OK	294 ms	405 bytes	7,502 bytes		Reflected	' ; drop table temp --
17	Fuzzed	200	OK	304 ms	405 bytes	7,635 bytes		Reflected	create user name identified by pass123 temporary tablespace temp default tablespace users;
16	Fuzzed	200	OK	303 ms	405 bytes	7,543 bytes		Reflected	create user name identified by 'pass123'
15	Fuzzed	200	OK	262 ms	406 bytes	7,560 bytes		Reflected	'; exec master..xp_cmdshell 'ping 10.10.1.2'--
14	Fuzzed	200	OK	289 ms	405 bytes	7,829 bytes			insert into users(login, password, level) values(char(0x70) + char(0x65) + char(0x74) + char(0x65) + char(0x72) + char(0x70) + cha
13	Fuzzed	200	OK	256 ms	405 bytes	7,545 bytes		Reflected	grant connect to name; grant resource to name;
12	Fuzzed	200	OK	321 ms	405 bytes	7,673 bytes		Reflected	insert into mysql.user (user, host, password) values ('name', 'localhost', password('pass123'))
11	Fuzzed	200	OK	300 ms	405 bytes	7,561 bytes		Reflected	exec sp_addsrvrolemember 'name' , 'sysadmin'
10	Fuzzed	200	OK	297 ms	405 bytes	7,545 bytes		Reflected	exec sp_addlogin 'name' , 'password'
9	Fuzzed	200	OK	282 ms	405 bytes	7,502 bytes		Reflected	' ; drop table temp --
8	Fuzzed	200	OK	281 ms	406 bytes	7,635 bytes		Reflected	create user name identified by pass123 temporary tablespace temp default tablespace users;
7	Fuzzed	200	OK	203 ms	405 bytes	7,543 bytes		Reflected	create user name identified by 'pass123'
6	Fuzzed	200	OK	208 ms	405 bytes	7,560 bytes		Reflected	'; exec master..xp_cmdshell 'ping 10.10.1.2'--
5	Fuzzed	200	OK	194 ms	405 bytes	7,560 bytes			"><script>alert('XSS')</script>
21	Fuzzed	200	OK	281 ms	405 bytes	7,673 bytes		Reflected	insert into mysql.user (user, host, password) values ('name', 'localhost', password('pass123'))
19	Fuzzed	200	OK	317 ms	406 bytes	7,545 bytes		Reflected	exec sp_addlogin 'name' , 'password'
22	Fuzzed	200	OK	302 ms	406 bytes	7,545 bytes		Reflected	grant connect to name; grant resource to name;
27	Fuzzed	200	OK	285 ms	405 bytes	7,569 bytes			<body/s/onload=x={doc:parent.document};x.doc.writeln(1)
23	Fuzzed	200	OK	302 ms	405 bytes	7,829 bytes			insert into users(login, password, level) values(char(0x70) + char(0x65) + char(0x74) + char(0x65) + char(0x72) + char(0x70) + cha
26	Fuzzed	200	OK	282 ms	405 bytes	7,577 bytes			<body onload=;a1={x:document};;;;;;;;;_=a1.x;_.write(1);;;;
25	Fuzzed	200	OK	287 ms	405 bytes	7,581 bytes			<body onload=a1={x:this.parent.document};a1.x.writeln(1);>
24	Fuzzed	200	OK	284 ms	405 bytes	7,625 bytes			<body onload=;a2={y:eval};a1={x:a2.y('al'+'ert')};;;;;;;;;_=a1.x;_(1);;;;
28	Fuzzed	200	OK	288 ms	405 bytes	7,579 bytes			<body/????$/onload=x={doc:parent[document]};x.doc.writeln(1)
29	Fuzzed	200	OK	282 ms	406 bytes	7,455 bytes		Reflected	|

Alerts 0 6 3 7 Main Proxy: localhost:8080

Current Scans 0 0 0 0 0 0 0 0

Vulnerability Found:

Using the **Injection** ruleset, OpenEMR shows debugging/failure information to the user. Therefore providing information to an attack that is not necessary. To fix this vulnerability, do not display the string in the dropdown box if there is an error. Log the error instead.

To replicate:

1. If not already running, start/run OpenEMR.

2. Open ZAP and create a new connection by selecting “Manual Explore”.

3. In “URL to explore”, enter: ‘http://localhost’. Leave “Enable HUD” unchecked, and then press “Launch Browser” to open your browser of choice. When the browser opens, if “Welcome to the ZAP HUD” appears, select “Continue to your target”.

4. Login with your admin account, unless changed, the credentials should be “admin” for the username, and “pass” for the password.

5. Select Reports, Then Visits. Then, Daily Report.

6. Select Submit.

7. Back in ZAP, in the left menu, navigate to the recent request:

Under “Sites”, click “http://localhost”.

Next, in the dropdown, select “interface”.

Next, in the dropdown, select “reports”.

Now, right click on “GET:daily_summary_report.php …” and select “Break…”

8. Select Save in the Add Breakpoint dialog.

9. Go back to OpenEMR and select the Submit button again.

10. In ZAP, go to the Body portion of the Break text box. Change &form_provider=& to &form_provider=23 or 1=1; --& .

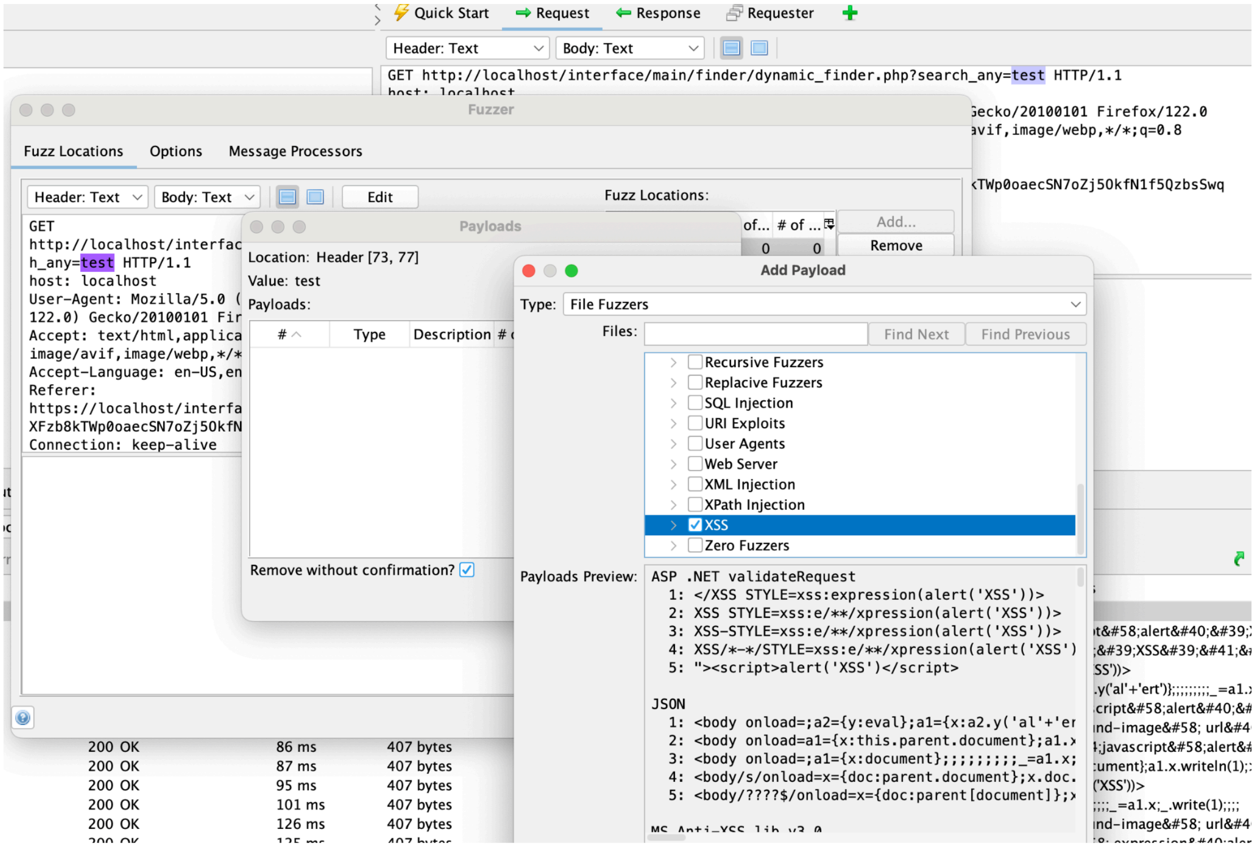
11. Then, select the Play button on ZAP until the button is disabled.

12. **RESULTS:** Back in OpenEMR, see that underneath the Provider dropdown, there is text that says Fix This!

Test 3:

Screenshot of ZAP rulesets used:

This test was for testing if user input is neutralized for XSS attacks. Since this test focused specifically on that, the payloads selected were for XSS.



```
GET http://localhost/interface/main/finder/dynamic_finder.php?search_any=test HTTP/1.1
host: localhost
```

```
GET
http://localhost/interfac
h_any=test HTTP/1.1
host: localhost
User-Agent: Mozilla/5.0 (
122.0) Gecko/20100101 Fir
Accept: text/html,applica
image/avif,image/webp,*/*
Accept-Language: en-US,en
Referer:
https://localhost/interfa
XFzb8kTWp0oaecSN7oZj5OkfN
Connection: keep-alive
```

Location: Header [73, 77]

Value: test

Payloads:

Remove without confirmation? ☑

Type: File Fuzzers

Files:

- □ Recursive Fuzzers
- □ Replacive Fuzzers
- □ SQL Injection
- □ URI Exploits
- □ User Agents
- □ Web Server
- □ XML Injection
- □ XPath Injection
- ☑ XSS
- □ Zero Fuzzers

Payloads Preview:

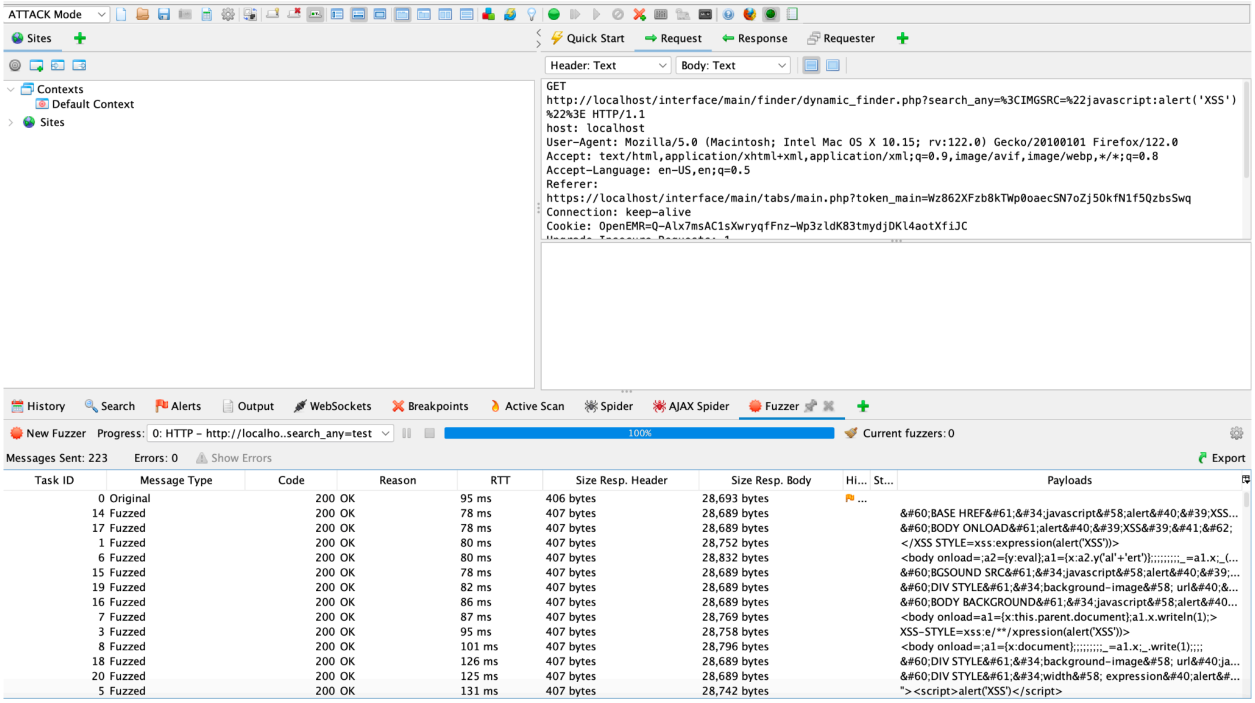
```
ASP .NET validateRequest
  1: </XSS STYLE=xss:expression(alert('XSS'))>
  2: XSS STYLE=xss:e/**/xpression(alert('XSS'))>
  3: XSS-STYLE=xss:e/**/xpression(alert('XSS'))>
  4: XSS/*-*/STYLE=xss:e/**/xpression(alert('XSS')
  5: "><script>alert('XSS')</script>

JSON
  1: <body onload=;a2={y:eval};a1={x:a2.y('al'+'er
  2: <body onload=a1={x:this.parent.document};a1.x
  3: <body onload=;a1={x:document};;;;;;;;;_=a1.x;
  4: <body/s/onload=x={doc:parent.document};x.doc.
  5: <body/????$/onload=x={doc:parent[document]};x

MS Anti-XSS lib v3.0
```

```
200 OK    86 ms    407 bytes
200 OK    87 ms    407 bytes
200 OK    95 ms    407 bytes
200 OK    101 ms   407 bytes
200 OK    126 ms   407 bytes
```

Screenshot of ZAP test results:



```
GET
http://localhost/interface/main/finder/dynamic_finder.php?search_any=%3CIMGSRC=%22javascript:alert('XSS')
%22%3E HTTP/1.1
host: localhost
User-Agent: Mozilla/5.0 (Macintosh; Intel Mac OS X 10.15; rv:122.0) Gecko/20100101 Firefox/122.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Referer:
https://localhost/interface/main/tabs/main.php?token_main=Wz862XFzb8kTWp0oaecSN7oZj5OkfN1f5QzbsSwq
Connection: keep-alive
Cookie: OpenEMR=Q-Alx7msAC1sXwryqfFnz-Wp3zldK83tmydjDKl4aotXfiJC
```

Progress: 0: HTTP – http://localho..search_any=test

Current fuzzers: 0

Messages Sent: 223 Errors: 0

Task ID	Message Type	Code	Reason	RTT	Size Resp. Header	Size Resp. Body	Hi...	St...	Payloads
0	Original	200	OK	95 ms	406 bytes	28,693 bytes	...		
14	Fuzzed	200	OK	78 ms	407 bytes	28,689 bytes			&#60;BASE HREF&#61;&#34;javascript&#58;alert&#40;&#39;XSS...
17	Fuzzed	200	OK	78 ms	407 bytes	28,689 bytes			&#60;BODY ONLOAD&#61;alert&#40;&#39;XSS&#39;&#41;&#62;
1	Fuzzed	200	OK	80 ms	407 bytes	28,752 bytes			</XSS STYLE=xss:expression(alert('XSS'))>
6	Fuzzed	200	OK	80 ms	407 bytes	28,832 bytes			<body onload=;a2={y:eval};a1={x:a2.y('al'+'ert')};;;;;;;;;_=a1.x;_(...
15	Fuzzed	200	OK	78 ms	407 bytes	28,689 bytes			&#60;BGSOUND SRC&#61;&#34;javascript&#58;alert&#40;&#39;...
19	Fuzzed	200	OK	82 ms	407 bytes	28,689 bytes			&#60;DIV STYLE&#61;&#34;background-image&#58; url&#40;&...
16	Fuzzed	200	OK	86 ms	407 bytes	28,689 bytes			&#60;BODY BACKGROUND&#61;&#34;javascript&#58;alert&#40...
7	Fuzzed	200	OK	87 ms	407 bytes	28,769 bytes			<body onload=a1={x:this.parent.document};a1.x.writeln(1);>
3	Fuzzed	200	OK	95 ms	407 bytes	28,758 bytes			XSS-STYLE=xss:e/**/xpression(alert('XSS'))>
8	Fuzzed	200	OK	101 ms	407 bytes	28,796 bytes			<body onload=;a1={x:document};;;;;;;;;_=a1.x;_.write(1);;;;
18	Fuzzed	200	OK	126 ms	407 bytes	28,689 bytes			&#60;DIV STYLE&#61;&#34;background-image&#58; url&#40;ja...
20	Fuzzed	200	OK	125 ms	407 bytes	28,689 bytes			&#60;DIV STYLE&#61;&#34;width&#58; expression&#40;alert&#...
5	Fuzzed	200	OK	131 ms	407 bytes	28,742 bytes			"><script>alert('XSS')</script>

Vulnerabilities Found:

ZAP reported that the code was 200 OK for all of the attacks, this indicates that the server is not neutralizing the code when it reaches the back end. In order to fix this vulnerability, the user-input text coming in needs to be either discarded or transformed into text that will not be processed as code by the server.

To replicate:

1) Using the ZAP browser, first execute a search in the search bar located at the top right of the screen. Enter ‘test’ and then press enter or the search icon to execute the search.
2) Back in ZAP, look for the request made in the history tab located at the bottom of the application. The request should show the following url: http://localhost/interface/main/finder/dynamic_finder.php?search_any=test
3) Double click that request which should open it in the request tab in ZAP. This is located at the top right of the ZAP screen.
4) In the first line, which should show:

   GET http://localhost/interface/main/finder/dynamic_finder.php?search_any=test HTTP/1.1

   Highlight the word ‘test’.

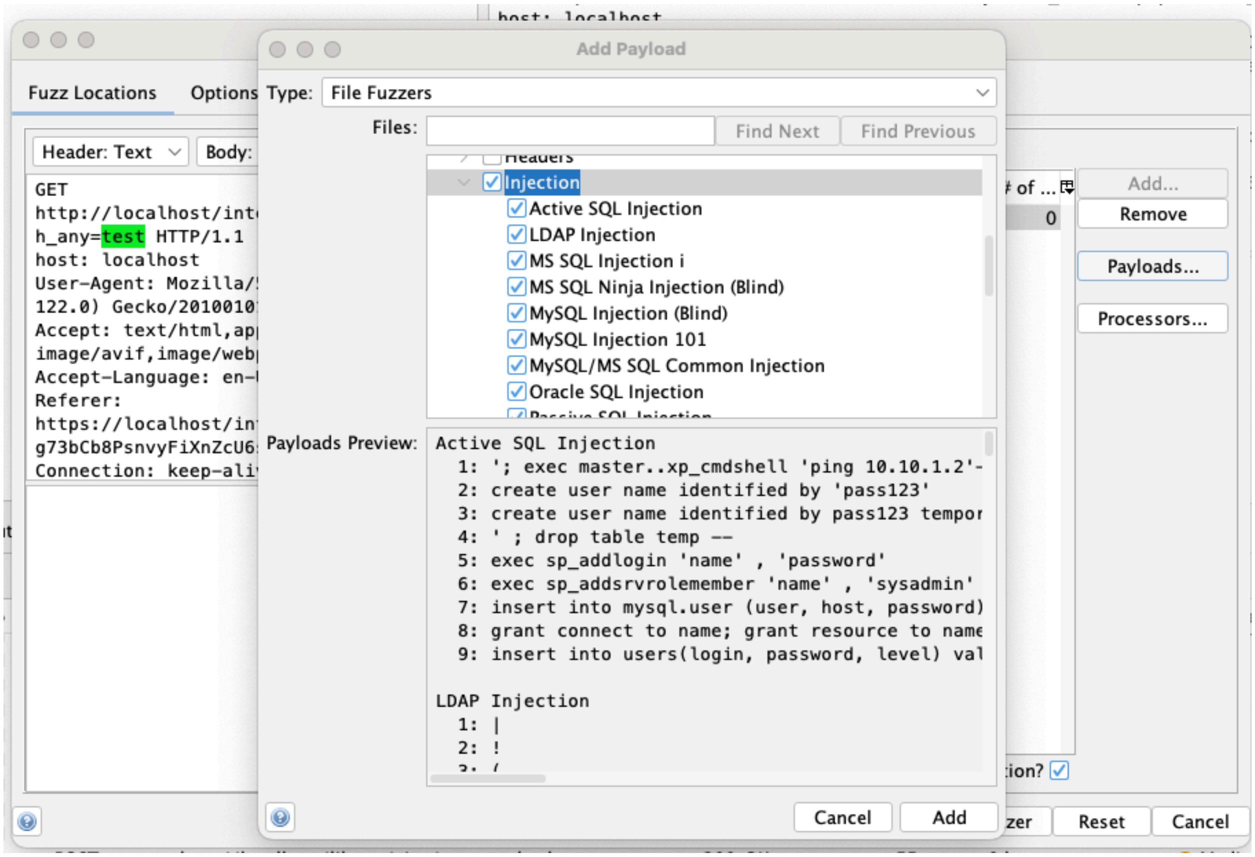
5) Then, right click on the highlighted ‘test’, and select Fuzz.
6) In the Fuzzer popup, select ‘Payloads…’, then on the Payloads popup, select ‘Add…’
7) In the Add Payload popup, change Type: from ‘Strings’ to ‘File Fuzzers’.
8) Next, press the arrow (it looks like ‘>’) located to the left of ‘jbrofuzz’.
9) Scroll down on the items located in ‘jbrofuzz’ and select the box next to ‘XSS’.
10) Select the ‘Add’ button at the bottom left of the ‘Add Payloads’ popup.

11) Select the ‘OK’ button at the bottom left of the ‘Payloads’ popup.
12) Finally, select ‘Start Fuzzer’ located at the bottom of the ‘Fuzzer’ popup.

Test 4:

Screenshot of ZAP rulesets used:

This test focused on injection, specifically with OS commands, which is why I selected Injection for the payload to use for this test.



```
GET
http://localhost/int
h_any=test HTTP/1.1
host: localhost
User-Agent: Mozilla/
122.0) Gecko/2010010
Accept: text/html,ap
image/avif,image/web
Accept-Language: en-
Referer:
https://localhost/in
g73bCb8PsnvyFiXnZcU6
Connection: keep-ali
```

Type: File Fuzzers

Files:

- ☐ Headers
- ☑ Injection
  - ☑ Active SQL Injection
  - ☑ LDAP Injection
  - ☑ MS SQL Injection i
  - ☑ MS SQL Ninja Injection (Blind)
  - ☑ MySQL Injection (Blind)
  - ☑ MySQL Injection 101
  - ☑ MySQL/MS SQL Common Injection
  - ☑ Oracle SQL Injection
  - ☑ Passive SQL Injection

Payloads Preview:

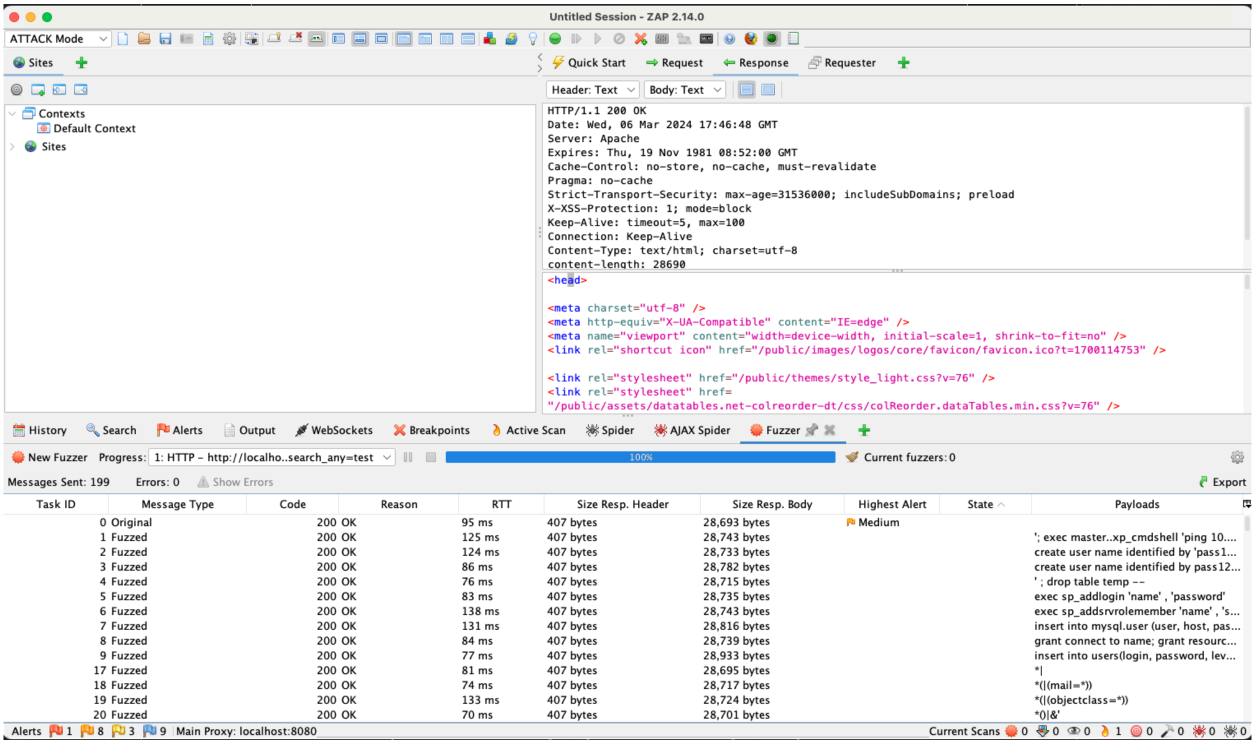
```
Active SQL Injection
  1: '; exec master..xp_cmdshell 'ping 10.10.1.2'-
  2: create user name identified by 'pass123'
  3: create user name identified by pass123 tempor
  4: ' ; drop table temp --
  5: exec sp_addlogin 'name' , 'password'
  6: exec sp_addsrvrolemember 'name' , 'sysadmin'
  7: insert into mysql.user (user, host, password)
  8: grant connect to name; grant resource to name
  9: insert into users(login, password, level) val

LDAP Injection
  1: |
  2: !
  3: (
```

of ... 0

ion? ☑

Screenshot of ZAP test results:



```
HTTP/1.1 200 OK
Date: Wed, 06 Mar 2024 17:46:48 GMT
Server: Apache
Expires: Thu, 19 Nov 1981 08:52:00 GMT
Cache-Control: no-store, no-cache, must-revalidate
Pragma: no-cache
Strict-Transport-Security: max-age=31536000; includeSubDomains; preload
X-XSS-Protection: 1; mode=block
Keep-Alive: timeout=5, max=100
Connection: Keep-Alive
Content-Type: text/html; charset=utf-8
content-length: 28690
```

```
<head>

<meta charset="utf-8" />
<meta http-equiv="X-UA-Compatible" content="IE=edge" />
<meta name="viewport" content="width=device-width, initial-scale=1, shrink-to-fit=no" />
<link rel="shortcut icon" href="/public/images/logos/core/favicon/favicon.ico?t=1700114753" />

<link rel="stylesheet" href="/public/themes/style_light.css?v=76" />
<link rel="stylesheet" href=
"/public/assets/datatables.net-colreorder-dt/css/colReorder.dataTables.min.css?v=76" />
```

Progress: 1: HTTP – http://localho..search_any=test

Current fuzzers: 0

Messages Sent: 199 Errors: 0

Task ID	Message Type	Code	Reason	RTT	Size Resp. Header	Size Resp. Body	Highest Alert	State	Payloads
0	Original	200 OK		95 ms	407 bytes	28,693 bytes	Medium		
1	Fuzzed	200 OK		125 ms	407 bytes	28,743 bytes			'; exec master..xp_cmdshell 'ping 10....
2	Fuzzed	200 OK		124 ms	407 bytes	28,733 bytes			create user name identified by 'pass1...
3	Fuzzed	200 OK		86 ms	407 bytes	28,782 bytes			create user name identified by pass12...
4	Fuzzed	200 OK		76 ms	407 bytes	28,715 bytes			' ; drop table temp --
5	Fuzzed	200 OK		83 ms	407 bytes	28,735 bytes			exec sp_addlogin 'name' , 'password'
6	Fuzzed	200 OK		138 ms	407 bytes	28,743 bytes			exec sp_addsrvrolemember 'name' , 's...
7	Fuzzed	200 OK		131 ms	407 bytes	28,816 bytes			insert into mysql.user (user, host, pas...
8	Fuzzed	200 OK		84 ms	407 bytes	28,739 bytes			grant connect to name; grant resourc...
9	Fuzzed	200 OK		77 ms	407 bytes	28,933 bytes			insert into users(login, password, lev...
17	Fuzzed	200 OK		81 ms	407 bytes	28,695 bytes			*|
18	Fuzzed	200 OK		74 ms	407 bytes	28,717 bytes			*(|(mail=*))
19	Fuzzed	200 OK		133 ms	407 bytes	28,724 bytes			*(|(objectclass=*))
20	Fuzzed	200 OK		70 ms	407 bytes	28,701 bytes			*()|&'

Main Proxy: localhost:8080

Vulnerabilities Found:

ZAP reported that the code was 200 OK for all of the attacks, this indicates that the server is not neutralizing the code when it reaches the back end. In order to fix this vulnerability, the user-input text coming in needs to be either discarded or transformed into text that will not be processed as code by the server.

To replicate:

1) Using the ZAP browser, first execute a search in the search bar located at the top right of the screen. Enter 'test' and then press enter or the search icon to execute the search.
2) Back in ZAP, look for the request made in the history tab located at the bottom of the application. The request should show the following url: http://localhost/interface/main/finder/dynamic_finder.php?search_any=test
3) Double click that request which should open it in the request tab in ZAP. This is located at the top right of the ZAP screen.
4) In the first line, which should show:

   GET http://localhost/interface/main/finder/dynamic_finder.php?search_any=test HTTP/1.1

   Highlight the word 'test'.

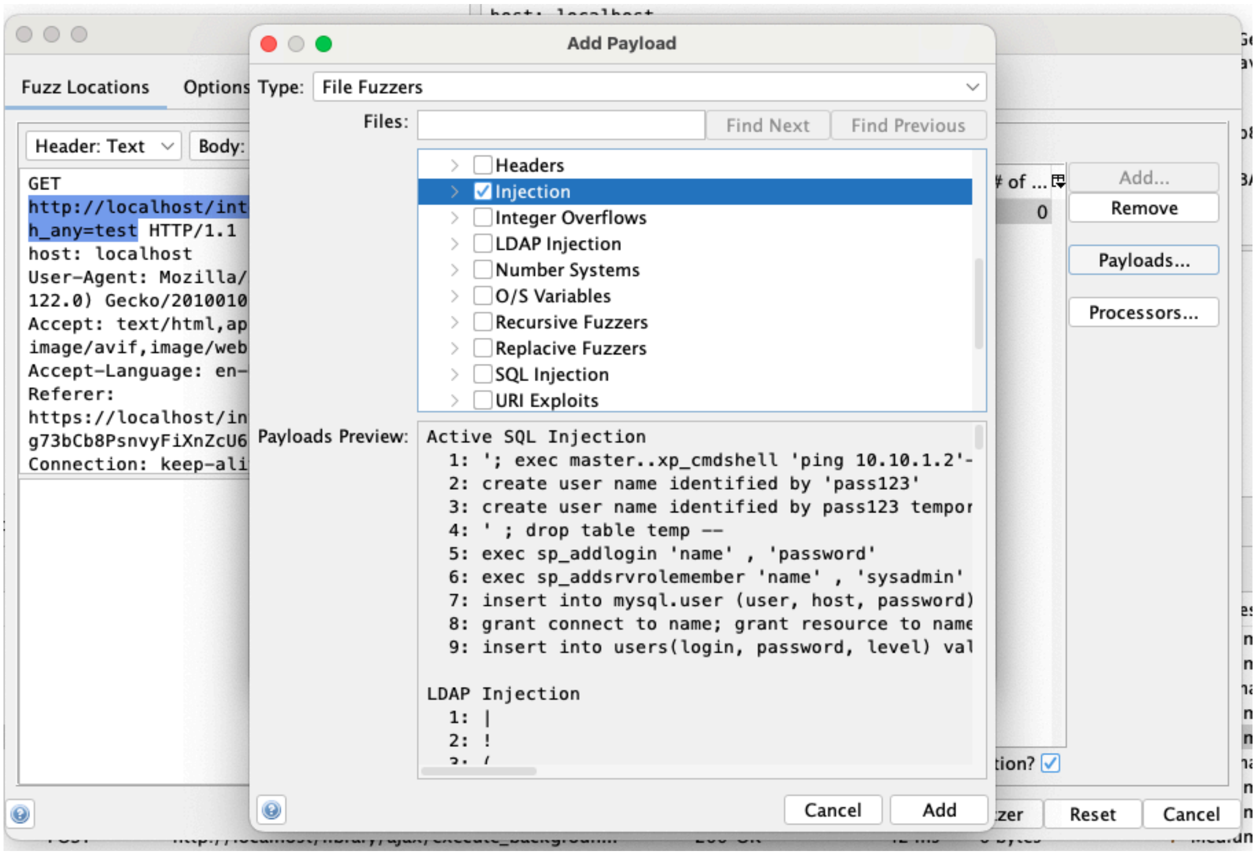
5) Then, right click on the highlighted 'test', and select Fuzz.
6) In the Fuzzer popup, select 'Payloads…', then on the Payloads popup, select 'Add…'
7) In the Add Payload popup, change Type: from 'Strings' to 'File Fuzzers'.
8) Next, press the arrow (it looks like '>') located to the left of 'jbrofuzz'.
9) Scroll down on the items located in 'jbrofuzz' and select the box next to 'Injection'.

10) Select the ‘Add’ button at the bottom left of the ‘Add Payloads’ popup.
11) Select the ‘OK’ button at the bottom left of the ‘Payloads’ popup.
12) Finally, select ‘Start Fuzzer’ located at the bottom of the ‘Fuzzer’ popup.

Test 5:

Screenshot of ZAP rulesets used:

This test focused on injection, specifically with injecting different URLs into the request, which is why I selected Injection for the payload to use for this test.



Type: File Fuzzers

```
GET
http://localhost/int
h_any=test HTTP/1.1
host: localhost
User-Agent: Mozilla/
122.0) Gecko/2010010
Accept: text/html,ap
image/avif,image/web
Accept-Language: en-
Referer:
https://localhost/in
g73bCb8PsnvyFiXnZcU6
Connection: keep-ali
```

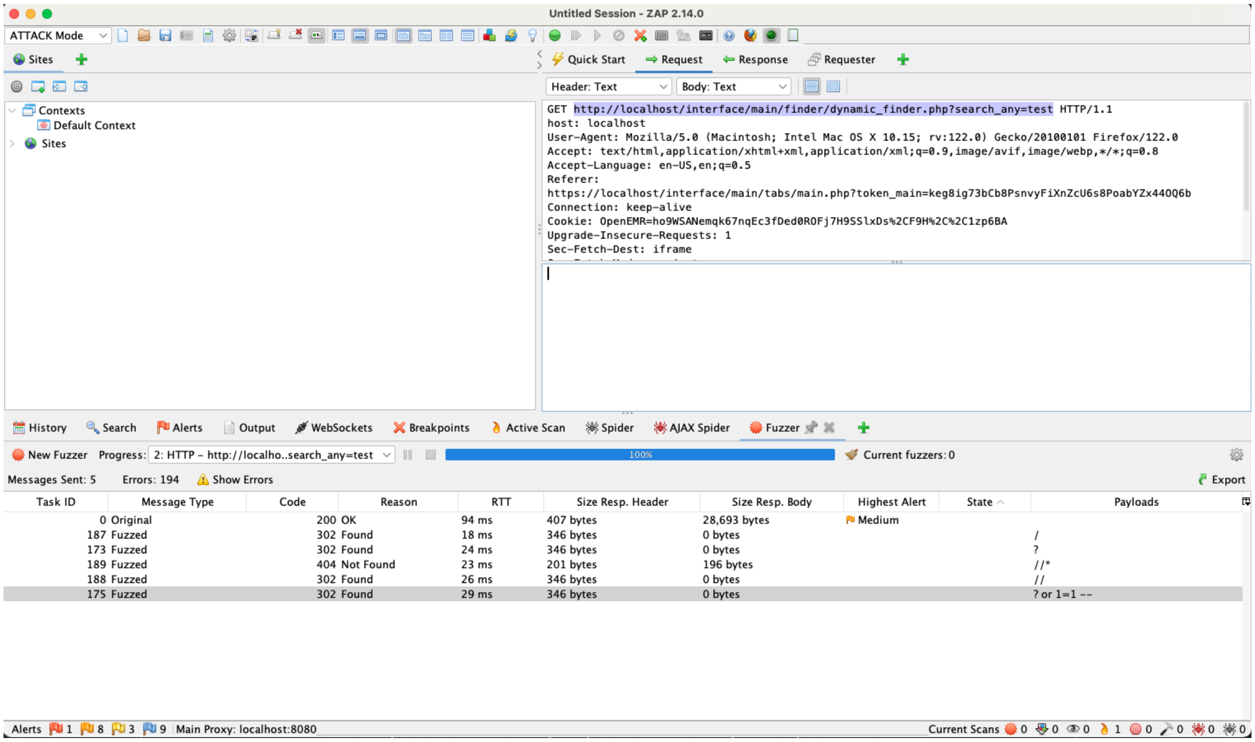
- □ Headers
- ☑ Injection
- □ Integer Overflows
- □ LDAP Injection
- □ Number Systems
- □ O/S Variables
- □ Recursive Fuzzers
- □ Replacive Fuzzers
- □ SQL Injection
- □ URI Exploits

Payloads Preview:

```
Active SQL Injection
  1: '; exec master..xp_cmdshell 'ping 10.10.1.2'-
  2: create user name identified by 'pass123'
  3: create user name identified by pass123 tempor
  4: ' ; drop table temp --
  5: exec sp_addlogin 'name' , 'password'
  6: exec sp_addsrvrolemember 'name' , 'sysadmin'
  7: insert into mysql.user (user, host, password)
  8: grant connect to name; grant resource to name
  9: insert into users(login, password, level) val

LDAP Injection
  1: |
  2: !
```

Screenshot of ZAP test results:



```
GET http://localhost/interface/main/finder/dynamic_finder.php?search_any=test HTTP/1.1
host: localhost
User-Agent: Mozilla/5.0 (Macintosh; Intel Mac OS X 10.15; rv:122.0) Gecko/20100101 Firefox/122.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Referer:
https://localhost/interface/main/tabs/main.php?token_main=keg8ig73bCb8PsnvyFiXnZcU6s8PoabYZx44OQ6b
Connection: keep-alive
Cookie: OpenEMR=ho9WSANemqk67nqEc3fDed0ROFj7H9SSlxDs%2CF9H%2C%2C1zp6BA
Upgrade-Insecure-Requests: 1
Sec-Fetch-Dest: iframe
```

Task ID	Message Type	Code	Reason	RTT	Size Resp. Header	Size Resp. Body	Highest Alert	State	Payloads
0	Original	200	OK	94 ms	407 bytes	28,693 bytes	Medium		
187	Fuzzed	302	Found	18 ms	346 bytes	0 bytes			/
173	Fuzzed	302	Found	24 ms	346 bytes	0 bytes			?
189	Fuzzed	404	Not Found	23 ms	201 bytes	196 bytes			//*
188	Fuzzed	302	Found	26 ms	346 bytes	0 bytes			//
175	Fuzzed	302	Found	29 ms	346 bytes	0 bytes			? or 1=1 --

Vulnerabilities Found:

ZAP reported that the code was 200 OK for most of the attacks. If you inspected the responses, you would see that the injection payloads were successful. The reason why this vulnerability was successful is that the server is not properly checking that the URL that it is redirecting or forwarding to is on an allow list (or one does not exist). So, by manipulating the URL of the GET request, we are able to get the server to load other websites or URLs within the application. This can be used to access untrusted or potentially dangerous sites.

To replicate:

1) Using the ZAP browser, first execute a search in the search bar located at the top right of the screen. Enter ‘test’ and then press enter or the search icon to execute the search.
2) Back in ZAP, look for the request made in the history tab located at the bottom of the application. The request should show the following url: http://localhost/interface/main/finder/dynamic_finder.php?search_any=test
3) Double click that request which should open it in the request tab in ZAP. This is located at the top right of the ZAP screen.
4) In the first line, which should show:

   GET http://localhost/interface/main/finder/dynamic_finder.php?search_any=test HTTP/1.1

   Highlight only the URL: ‘http://localhost/interface/main/finder/dynamic_finder.php?search_any=test’.

5) Then, right click on the highlighted URL, and select Fuzz.
6) In the Fuzzer popup, select ‘Payloads…’, then on the Payloads popup, select ‘Add…’
7) In the Add Payload popup, change Type: from ‘Strings’ to ‘File Fuzzers’.

8) Next, press the arrow (it looks like ‘>’) located to the left of ‘jbrofuzz’.
9) Scroll down on the items located in ‘jbrofuzz’ and select the box next to ‘Injection’.
10) Select the ‘Add’ button at the bottom left of the ‘Add Payloads’ popup.
11) Select the ‘OK’ button at the bottom left of the ‘Payloads’ popup.
12) Finally, select ‘Start Fuzzer’ located at the bottom of the ‘Fuzzer’ popup.

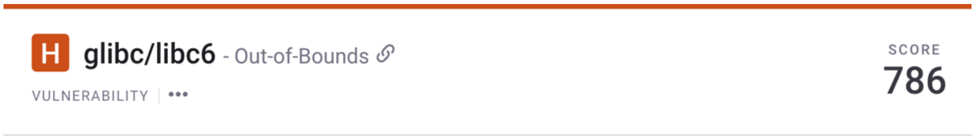
2. Vulnerable Dependencies:

1) Snyk:

- Result: The number of total vulnerable dependencies are 34.

1.

- CVE-2017-1000409
- CWE: 119
- Direct Dependency
- Safer Version: glibc **2.26**



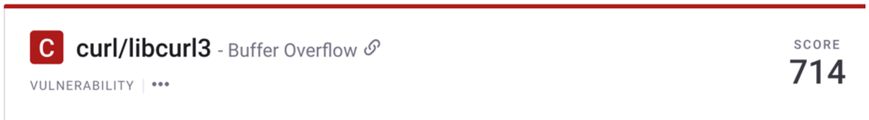
H **glibc/libc6** - Out-of-Bounds

VULNERABILITY | •••

SCORE
786

2.

- CVE-2019-5482
- CWE:120
- Direct Dependency
- Safer Version: CURL version **7.65.3** and later.



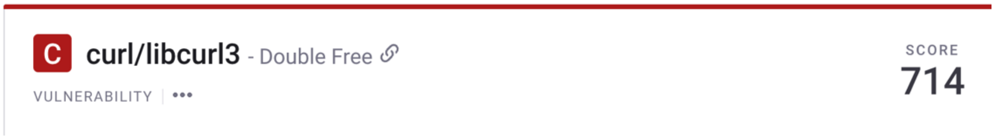
C **curl/libcurl3** - Buffer Overflow

VULNERABILITY | •••

SCORE
714

3.

- CVE-2019-5481
- CWE:415
- Direct Dependency
- Safer Version: CURL version **7.78.0** and later.



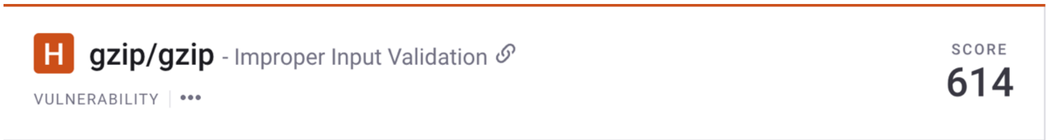
C **curl/libcurl3** - Double Free

VULNERABILITY | •••

SCORE
714

4.

- CVE-2022-1271
- CWE:20
- Transitive Dependency
- Safer Version: GZIP Version **1.12**



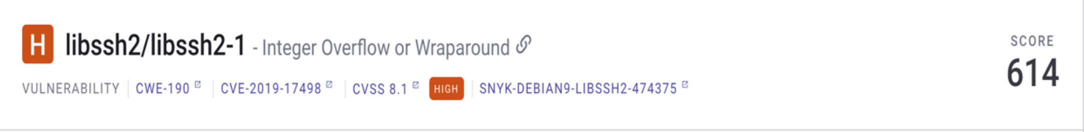
H gzip/gzip - Improper Input Validation

VULNERABILITY | •••

SCORE 614

5.

- CVE-2019-17498
- CWE:190
- Direct Dependency
- Safer Version: libssh2 Version 1.9.1



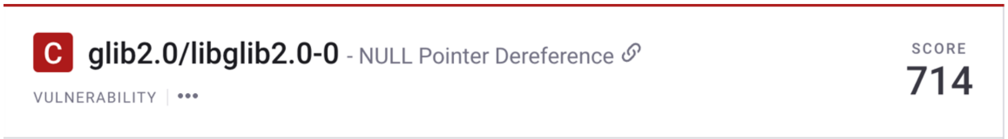
H libssh2/libssh2-1 - Integer Overflow or Wraparound

VULNERABILITY | CWE-190 | CVE-2019-17498 | CVSS 8.1 HIGH | SNYK-DEBIAN9-LIBSSH2-474375

SCORE 614

6.

- CVE-2018-16428
- CWE:476
- Direct Dependency
- Safer Version: GNOME GLib **2.56.1** and later.



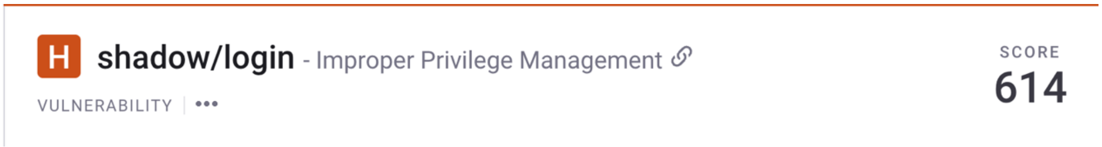
C glib2.0/libglib2.0-0 - NULL Pointer Dereference

VULNERABILITY | •••

SCORE 714

7.

- CVE-2017-20002
- CWE:269
- Direct Dependency
- Safer Version: version **greater than or equal to 4.5-1**


H **shadow/login** - Improper Privilege Management

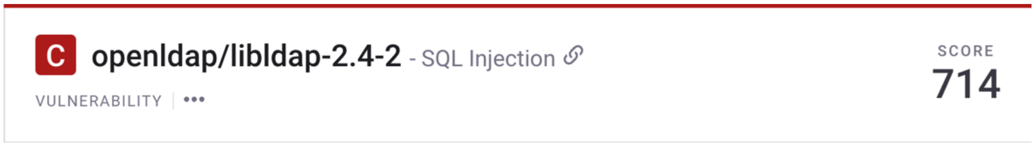
VULNERABILITY | •••

SCORE

614

8.

- CVE-2022-29155
- CWE:89
- Both Direct and Transitive Dependencies
- Safer Version: **OpenLDAP 2.5.12** or **2.6.2** (or later)


C **openldap/libldap-2.4-2** - SQL Injection

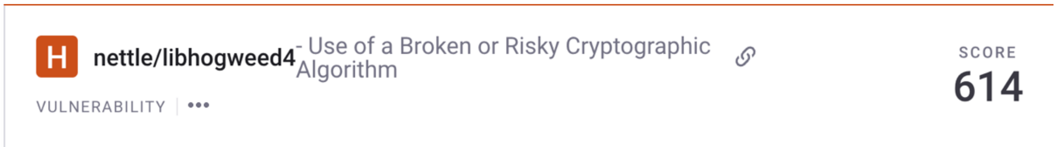
VULNERABILITY | •••

SCORE

714

9.

- CVE-2021-20305
- CWE:327
- Both Direct and Transitive Dependencies
- Safer Version: Nettle version **3.7.2** or later


H **nettle/libhogweed4**- Use of a Broken or Risky Cryptographic Algorithm

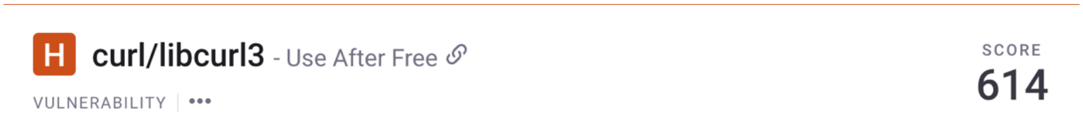
VULNERABILITY | •••

SCORE

614

10.

- CVE-2020-8231
- CWE:416
- Transitive Dependencies
- Safer Version: latest patched version of libcurl**.**
-


H **curl/libcurl3** - Use After Free

VULNERABILITY | •••

SCORE

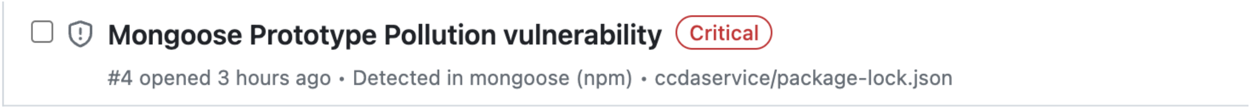
614

2) **GitHub's checker:**

- Result: The number of total vulnerable dependencies are 25.

1.

- CVE-2023-3696
- CWE:1321
- Direct Dependencies
- Safer Version: mongoose version 5.13.20 and later

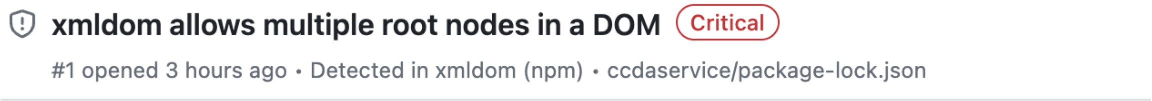


□ **Mongoose Prototype Pollution vulnerability** Critical

#4 opened 3 hours ago • Detected in mongoose (npm) • ccdaservice/package-lock.json

2.

- CVE-2022-39353
- CWE:20, 1288
- Direct Dependencies
- Safer Version: Version **0.7.7**, **0.8.4,** or >=**0.9.0-beta.**

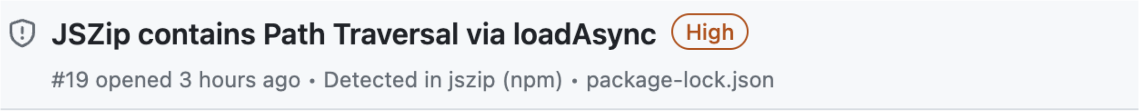


xmldom allows multiple root nodes in a DOM Critical

#1 opened 3 hours ago • Detected in xmldom (npm) • ccdaservice/package-lock.json

3.

- CVE-2022-48285
- CWE: 22
- Both Direct and Transitive Dependencies
- Safer Version: JSZip **3.8.0**

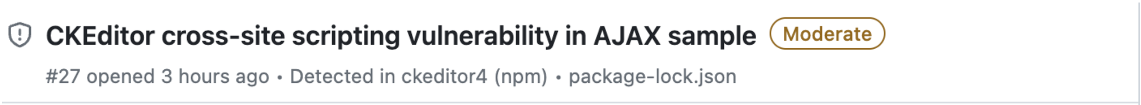


JSZip contains Path Traversal via loadAsync High

#19 opened 3 hours ago • Detected in jszip (npm) • package-lock.json

4.

- CVE-2023-4771
- CWE: 79
- Direct Dependencies
- Safer Version: CKeditor4 Version **4.24.0-lts**

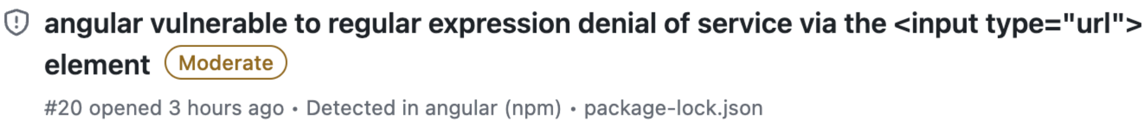


CKEditor cross-site scripting vulnerability in AJAX sample Moderate

#27 opened 3 hours ago • Detected in ckeditor4 (npm) • package-lock.json

5.

- CVE-2023-26118
- CWE: 1333
- Both Direct and Transitive Dependencies
- Safer Version: Unknown patch version

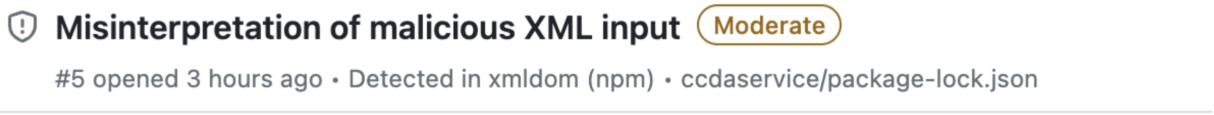


angular vulnerable to regular expression denial of service via the <input type="url"> element Moderate

#20 opened 3 hours ago • Detected in angular (npm) • package-lock.json

6.

- CVE-2021-32796
- CWE: 116
- Transitive Dependencies
- Safer Version: Unknown patch version

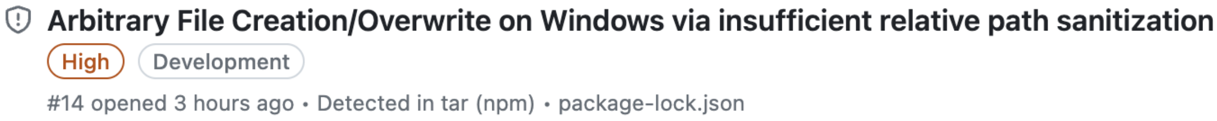


Misinterpretation of malicious XML input Moderate

#5 opened 3 hours ago • Detected in xmldom (npm) • ccdaservice/package-lock.json

7.

- CVE-2021-37713
- CWE: 22
- Direct Dependencies
- Safer Version: tar Version **4.4.18**

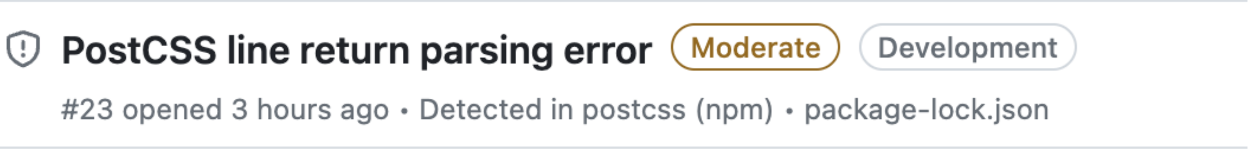


Arbitrary File Creation/Overwrite on Windows via insufficient relative path sanitization High Development

#14 opened 3 hours ago • Detected in tar (npm) • package-lock.json

8.

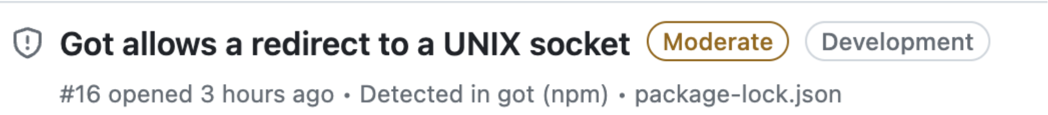
- CVE-2023-44270
- CWE: 74, 144
- Direct Dependencies
- Safer Version: Postcss Version **8.4.31**


PostCSS line return parsing error

#23 opened 3 hours ago • Detected in postcss (npm) • package-lock.json

9.

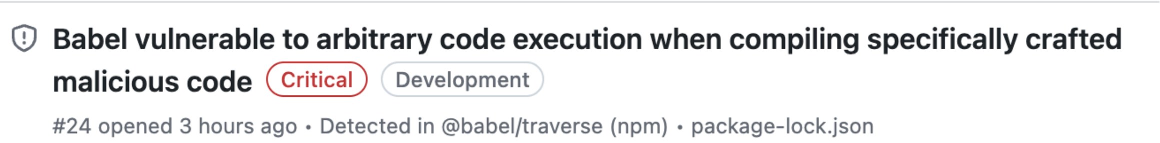
- CVE-2022-33987
- CWE: NO CWEs
- Both Direct and Transitive Dependencies
- Safer Version: got Version **11.8.5.**


Got allows a redirect to a UNIX socket

#16 opened 3 hours ago • Detected in got (npm) • package-lock.json

10.

- CVE-2023-45133
- CWE: 184, 697
- Direct Dependencies
- Safer Version: @babel/traverseVersion **7.23.2.**


Babel vulnerable to arbitrary code execution when compiling specifically crafted malicious code

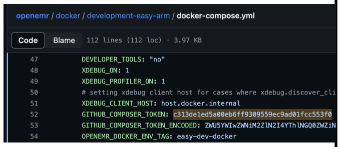
#24 opened 3 hours ago • Detected in @babel/traverse (npm) • package-lock.json

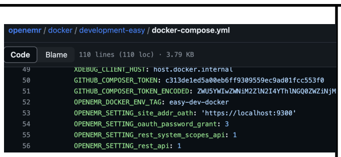
Snyk has shown more vulnerabilities than **GitHub's Checker**, while both tools aim to address vulnerable dependencies, their underlying mechanisms and focus areas differ. Snyk provides a holistic view and prioritizes ease of fixing, whereas GitHub's checker operates based on manifest files and may not emphasize the same level of user-friendly fixes. Understanding these distinctions can help you choose the most suitable tool for your specific needs.

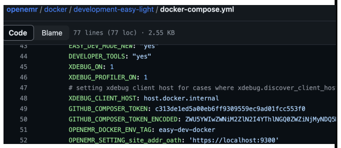
Snyk provides a holistic security solution with community support and emphasizes ease of fixing. **GitHub's checker** focuses on manifest-based analysis and integrates seamlessly with GitHub. **Bomber** targets SBOM scanning for security vulnerabilities.

3. Secret Detection:

Gitleaks

Secret Types	Exposed Secret	Screenshot	Tool-generated output
generic-api-key	ZWU5YWIwZWNiM2ZlN2I4YThlNGQ0ZWZiNjMyNDQ5MjFkZTJhMTY2OQo=	 openemr / docker / development-easy-arm / docker-compose.yml Code Blame 112 lines (112 loc) · 3.97 KB 47 DEVELOPER_TOOLS: "no" 48 XDEBUG_ON: 1 49 XDEBUG_PROFILER_ON: 1 50 # setting xdebug client host for cases where xdebug.discover_cli 51 XDEBUG_CLIENT_HOST: host.docker.internal 52 GITHUB_COMPOSER_TOKEN: c313de1ed5a00eb6ff9309559ec9ad01fcc553f0 53 GITHUB_COMPOSER_TOKEN_ENCODED: ZWU5YWIwZWNiM2ZlN2I4YThlNGQ0ZWZiN 54 OPENEMR_DOCKER_ENV_TAG: easy-dev-docker	Finding: GITHUB_COMPOSER_TOKEN_ENCODED: ZWU5YWIwZWNiM2ZlN2I4YThlNGQ0ZWZiNjMyNDQ5MjFkZTJhMTY2OQo= Secret: ZWU5YWIwZWNiM2ZlN2I4YThlNGQ0ZWZiNjMyNDQ5MjFkZTJhMTY2OQo= RuleID: generic-api-key Entropy: 4.483445 File: docker/development-easy-arm/docker-compose.yml Line: 52 Commit: fe61175c8f0cc33552004ddc122b0722043f0d29 Author: Brady Miller Email: brady.g.miller@gmail.com Date: 2021-03-20T23:16:32Z Fingerprint: fe61175c8f0cc33552004ddc122b0722043f0d29:docker/development-easy-arm64/docker-compose.yml:generic-api-key:52

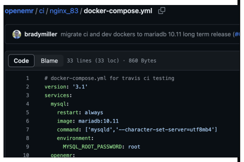
generic-api-key	ZWU5YWIwZWNiM2ZlN2I4YThlNGQ0ZWZiNjMyNDQ5MjFkZTJhMTY2OQo=	
		openemr / docker / development-easy / docker-compose.yml
		Code Blame 110 lines (110 loc) · 3.79 KB
		`49 XDEBUG_CLIENT_HOST: host.docker.internal`
		`50 GITHUB_COMPOSER_TOKEN: c313de1ed5a00eb6ff9309559ec9ad01fcc553f0`
		`51 GITHUB_COMPOSER_TOKEN_ENCODED: ZWU5YWIwZWNiM2ZlN2I4YThlNGQ0ZWZiNjM`
		`52 OPENEMR_DOCKER_ENV_TAG: easy-dev-docker`
		`53 OPENEMR_SETTING_site_addr_oath: 'https://localhost:9300'`
		`54 OPENEMR_SETTING_oauth_password_grant: 3`
		`55 OPENEMR_SETTING_rest_system_scopes_api: 1`
		`56 OPENEMR_SETTING_rest_api: 1`
			Finding: GITHUB_COMPOSER_TOKEN_ENCODED: ZWU5YWIwZWNiM2ZlN2I4YThlNGQ0ZWZiNjMyNDQ5MjFkZTJhMTY2OQo=
			Secret: ZWU5YWIwZWNiM2ZlN2I4YThlNGQ0ZWZiNjMyNDQ5MjFkZTJhMTY2OQo=
			RuleID: generic-api-key
			Entropy: 4.483445
			File: docker/development-easy/docker-compose.yml
			Line: 50
			Commit: fe61175c8f0cc33552004ddc122b0722043f0d29
			Author: Brady Miller
			Email: brady.g.miller@gmail.com
			Date: 2021-03-20T23:16:32Z
			Fingerprint: fe61175c8f0cc33552004ddc122b0722043f0d29:docker/development-easy/docker-compose.yml:generic-api-key:50

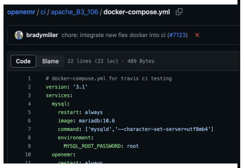
generic-api-key	ZWU5YWIwZWNiM2ZlN2I4YThlNGQ0ZWZiNjMyNDQ5MjFkZTJhMTY2OQo=	
		openemr / docker / development-easy-light / docker-compose.yml
		Code Blame 77 lines (77 loc) · 2.55 KB
		`43 EASY_DEV_MODE_NEW: "yes"`
		`44 DEVELOPER_TOOLS: "yes"`
		`45 XDEBUG_ON: 1`
		`46 XDEBUG_PROFILER_ON: 1`
		`47 # setting xdebug client host for cases where xdebug.discover_client_hos`
		`48 XDEBUG_CLIENT_HOST: host.docker.internal`
		`49 GITHUB_COMPOSER_TOKEN: c313de1ed5a00eb6ff9309559ec9ad01fcc553f0`
		`50 GITHUB_COMPOSER_TOKEN_ENCODED: ZWU5YWIwZWNiM2ZlN2I4YThlNGQ0ZWZiNjMyNDQ5`
		`51 OPENEMR_DOCKER_ENV_TAG: easy-dev-docker`
		`52 OPENEMR_SETTING_site_addr_oath: 'https://localhost:9300'`
			Finding: GITHUB_COMPOSER_TOKEN_ENCODED: ZWU5YWIwZWNiM2ZlN2I4YThlNGQ0ZWZiNjMyNDQ5MjFkZTJhMTY2OQo=
			Secret: ZWU5YWIwZWNiM2ZlN2I4YThlNGQ0ZWZiNjMyNDQ5MjFkZTJhMTY2OQo=
			RuleID: generic-api-key
			Entropy: 4.483445
			File: docker/development-easy-light/docker-compose.yml
			Line: 49
			Commit: fe61175c8f0cc33552004ddc122b0722043f0d29
			Author: Brady Miller
			Email: brady.g.miller@gmail.com
			Date: 2021-03-20T23:16:32Z
			Fingerprint: fe61175c8f0cc33552004ddc122b0722043f0d29:docker/development-easy-light/docker-compose.yml:generic-api-key:49

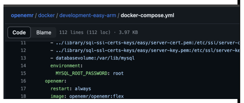
Reflection: This tool produced several false positives by misidentifying content in files that contain encoded text. The tool associated long strings with random characters as secrets. Also this tool detected mock passwords that were used in unit tests and example tokens that were provided in the readme which describes how to use the OpenEMRs API. If a variable in the code has a name that suggests that it may hold sensitive information, the tool tends to report that as well. Overall, three true positives were found in three different docker .yml configuration files.

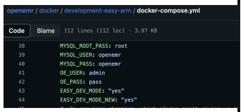
whispers

Secret Types	Exposed Secret	Screenshot	Tool-generated output

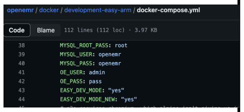
password	root	
		openemr / ci / nginx_83 / docker-compose.yml
		bradymiller migrate ci and dev dockers to mariadb 10.11 long term release (#
		Code Blame 33 lines (33 loc) · 860 Bytes
		# docker-compose.yml for travis ci testing
		version: '3.1'
		services:
		mysql:
		restart: always
		image: mariadb:10.11
		command: ['mysqld','--character-set-server=utf8mb4']
		environment:
		MYSQL_ROOT_PASSWORD: root
		openemr:
			{
			"key": "MYSQL_ROOT_PASSWORD",
			"value": "root",
			"file": "openemr/ci/nginx_83/docker-compose.yml",
			"line": 9,
			"rule_id": "password",
			"message": "Password",
			"severity": "High"
			}

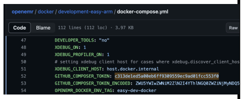
password	root	
		openemr / ci / apache_83_106 / docker-compose.yml
		bradymiller chore: integrate new flex docker into ci (#7123) ✕
		Code Blame 22 lines (22 loc) · 489 Bytes
		# docker-compose.yml for travis ci testing
		version: '3.1'
		services:
		mysql:
		restart: always
		image: mariadb:10.6
		command: ['mysqld','--character-set-server=utf8mb4']
		environment:
		MYSQL_ROOT_PASSWORD: root
		openemr:
		restart: always
			{
			"key": "MYSQL_ROOT_PASSWORD",
			"value": "root",
			"file": "openemr/ci/apache_83_106/docker-compose.yml",
			"line": 9,
			"rule_id": "password",
			"message": "Password",
			"severity": "High"
			}

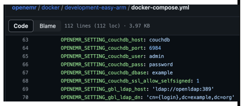
password	root	
		openemr / docker / development-easy-arm / docker-compose.yml
		Code Blame 112 lines (112 loc) · 3.97 KB
		- ../library/sql-ssl-certs-keys/easy/server-cert.pem:/etc/ssl/server-c
		- ../library/sql-ssl-certs-keys/easy/server-key.pem:/etc/ssl/server-ke
		- databasevolume:/var/lib/mysql
		environment:
		MYSQL_ROOT_PASSWORD: root
		openemr:
		restart: always
		image: openemr/openemr:flex
			{
			"key": "MYSQL_ROOT_PASSWORD",
			"value": "root",
			"file": "openemr/docker/development-easy-arm/docker-compose.yml",
			"line": 15,
			"rule_id": "password",
			"message": "Password",
			"severity": "High"
			}

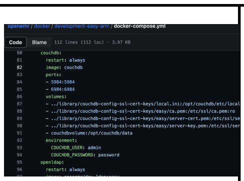
password	openemr	 openemr / docker / development-easy-arm / docker-compose.yml Code Blame 112 lines (112 loc) · 3.97 KB MYSQL_ROOT_PASS: root MYSQL_USER: openemr MYSQL_PASS: openemr OE_USER: admin OE_PASS: pass EASY_DEV_MODE: "yes" EASY_DEV_MODE_NEW: "yes"	{ "key": "MYSQL_PASS", "value": "openemr", "file": "openemr/docker/development-easy-arm/docker-compose.yml", "line": 40, "rule_id": "password", "message": "Password",

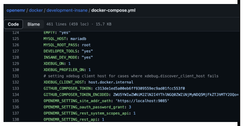
			"severity": "High" }

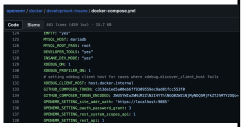
password	pass	
		openemr / docker / development-easy-arm / docker-compose.yml
		Code Blame 112 lines (112 loc) · 3.97 KB
		`38 MYSQL_ROOT_PASS: root`
		`39 MYSQL_USER: openemr`
		`40 MYSQL_PASS: openemr`
		`41 OE_USER: admin`
		`42 OE_PASS: pass`
		`43 EASY_DEV_MODE: "yes"`
		`44 EASY_DEV_MODE_NEW: "yes"`
			{
			"key": "OE_PASS",
			"value": "pass",
			"file": "openemr/docker/development-easy-arm/docker-compose.yml",
			"line": 42,
			"rule_id": "password",
			"message": "Password",
			"severity": "High"
			}

apikey	c313de1ed5a00eb6ff9309559ec9ad01fcc553f0	
		openemr / docker / development-easy-arm / docker-compose.yml
		Code Blame 112 lines (112 loc) · 3.97 KB Raw
		`47 DEVELOPER_TOOLS: "no"`
		`48 XDEBUG_ON: 1`
		`49 XDEBUG_PROFILER_ON: 1`
		`50 # setting xdebug client host for cases where xdebug.discover_client_hos`
		`51 XDEBUG_CLIENT_HOST: host.docker.internal`
		`52 GITHUB_COMPOSER_TOKEN: c313de1ed5a00eb6ff9309559ec9ad01fcc553f0`
		`53 GITHUB_COMPOSER_TOKEN_ENCODED: ZWU5YWIwZWNiM2ZlN2I4YThlNGQ0ZWZiNjMyNDQ5`
		`54 OPENEMR_DOCKER_ENV_TAG: easy-dev-docker`
			{
			"key": "GITHUB_COMPOSER_TOKEN",
			"value": "c313de1ed5a00eb6ff9309559ec9ad01fcc553f0",
			"file": "openemr/docker/development-easy-arm/docker-compose.yml",
			"line": 52,
			"rule_id": "apikey",
			"message": "API key",
			"severity": "Medium"
			}

password	password	
		openemr / docker / development-easy-arm / docker-compose.yml
		Code Blame 112 lines (112 loc) · 3.97 KB
		`63 OPENEMR_SETTING_couchdb_host: couchdb`
		`64 OPENEMR_SETTING_couchdb_port: 6984`
		`65 OPENEMR_SETTING_couchdb_user: admin`
		`66 OPENEMR_SETTING_couchdb_pass: password`
		`67 OPENEMR_SETTING_couchdb_dbase: example`
		`68 OPENEMR_SETTING_couchdb_ssl_allow_selfsigned: 1`
		`69 OPENEMR_SETTING_gbl_ldap_host: 'ldap://openldap:389'`
		`70 OPENEMR_SETTING_gbl_ldap_dn: 'cn={login},dc=example,dc=org'`
			{
			"key": "OPENEMR_SETTING_couchdb_pass",
			"value": "password",
			"file": "openemr/docker/development-easy-arm/docker-compose.yml",
			"line": 66,
			"rule_id": "password",
			"message": "Password",
			"severity": "High"
			}

password	password	
		openemr / docker / development-easy-arm / docker-compose.yml
		Code Blame 112 lines (112 loc) · 3.97 KB
		80 couchdb:
		81 restart: always
		82 image: couchdb
		83 ports:
		84 - 5984:5984
		85 - 6984:6984
		86 volumes:
		87 - ../library/couchdb-config-ssl-cert-keys/local.ini:/opt/couchdb/etc/local
		88 - ../library/couchdb-config-ssl-cert-keys/easy/ca.pem:/etc/ssl/ca.pem:ro
		89 - ../library/couchdb-config-ssl-cert-keys/easy/server-cert.pem:/etc/ssl/se
		90 - ../library/couchdb-config-ssl-cert-keys/easy/server-key.pem:/etc/ssl/ser
		91 - couchdbvolume:/opt/couchdb/data
		92 environment:
		93 COUCHDB_USER: admin
		94 COUCHDB_PASSWORD: password
		95 openldap:
		96 restart: always
			{
			"key": "COUCHDB_PASSWORD",
			"value": "password",
			"file": "openemr/docker/development-easy-arm/docker-compose.yml",
			"line": 94,
			"rule_id": "password",
			"message": "Password",
			"severity": "High"
			}

password	c313de1ed5a00eb6ff9309559ec9ad01fcc553f0	
		openemr / docker / development-insane / docker-compose.yml
		Code Blame 461 lines (459 loc) · 15.7 KB
		124 EMPTY: "yes"
		125 MYSQL_HOST: mariadb
		126 MYSQL_ROOT_PASS: root
		127 DEVELOPER_TOOLS: "yes"
		128 INSANE_DEV_MODE: "yes"
		129 XDEBUG_ON: 1
		130 XDEBUG_PROFILER_ON: 1
		131 # setting xdebug client host for cases where xdebug.discover_client_host fails
		132 XDEBUG_CLIENT_HOST: host.docker.internal
		133 GITHUB_COMPOSER_TOKEN: c313de1ed5a00eb6ff9309559ec9ad01fcc553f0
		134 GITHUB_COMPOSER_TOKEN_ENCODED: ZWU5YWIwZWNiM2ZlN2I4YThlNGQ0ZWZiNjMyNDQ5MjFkZTJhMTY2OQo=
		135 OPENEMR_SETTING_site_addr_oath: 'https://localhost:9085'
		136 OPENEMR_SETTING_oauth_password_grant: 3
		137 OPENEMR_SETTING_rest_system_scopes_api: 1
		138 OPENEMR_SETTING_rest_api: 1
			{
			"key": "MYSQL_ROOT_PASS",
			"value": "root",
			"file": "openemr/docker/development-insane/docker-compose.yml",
			"line": 126,
			"rule_id": "password",
			"message": "Password",
			"severity": "High"
			}

apikey	c313de1ed5a00eb6ff9309559ec9ad01fcc553f0	
		openemr / docker / development-insane / docker-compose.yml
		Code Blame 461 lines (459 loc) · 15.7 KB
		124 EMPTY: "yes"
		125 MYSQL_HOST: mariadb
		126 MYSQL_ROOT_PASS: root
		127 DEVELOPER_TOOLS: "yes"
		128 INSANE_DEV_MODE: "yes"
		129 XDEBUG_ON: 1
		130 XDEBUG_PROFILER_ON: 1
		131 # setting xdebug client host for cases where xdebug.discover_client_host fails
		132 XDEBUG_CLIENT_HOST: host.docker.internal
		133 GITHUB_COMPOSER_TOKEN: c313de1ed5a00eb6ff9309559ec9ad01fcc553f0
		134 GITHUB_COMPOSER_TOKEN_ENCODED: ZWU5YWIwZWNiM2ZlN2I4YThlNGQ0ZWZiNjMyNDQ5MjFkZTJhMTY2OQo=
		135 OPENEMR_SETTING_site_addr_oath: 'https://localhost:9085'
		136 OPENEMR_SETTING_oauth_password_grant: 3
		137 OPENEMR_SETTING_rest_system_scopes_api: 1
		138 OPENEMR_SETTING_rest_api: 1
			{
			"key": "GITHUB_COMPOSER_TOKEN",
			"value": "c313de1ed5a00eb6ff9309559ec9ad01fcc553f0",
			"file": "openemr/docker/development-insane/docker-compose.yml",
			"line": 133,
			"rule_id": "apikey",
			"message": "API key",
			"severity": "Medium"
			}

Reflection: This tool did a decent job finding passwords and keys located in configuration files. There was a noticeable amount (10+) of true positives. On the other hand, it also tended to misidentify regular properties in standard json configuration files like package-lock.json. Additional false positives were also identified in html files which were likely mistaken.

Comparison Report

Both tools seem to have different strengths and weaknesses. While the majority of its reported secrets were false positives, Gitleaks did provide a more in depth analysis of the entire OpenEMR codebase. It attempted to discover secrets by going through the entire repo. Whisper identified several true positives, but it neglected to search through the code and mostly provided results that were from configuration files. **Discussion:** What was interesting about both tools was that they did a decent job of identifying secrets stored in .yml files. Github tokens were identified by both tools and most of the time, the value and line number provided in the output from the tools were correct. Since Whisper was designed to find secrets specifically in config files, it did a better job at identifying the tokens in yml files than Gitleaks. While Gitleaks did not identify all the secrets that Whisper found, Gitleaks still managed to identify potential security concerns in the actual code.

How OpenEMR handles Secrets

In code, OpenEMR uses configuration files to set sensitive data. However, in many of the configuration files, secrets are plainly embedded. These plain secrets are primarily found in docker yaml and ci/cd files. It would be safer for OpenEMR to store secrets in environment variables and then reference the values in the configuration files. The discovered secrets protect private github resources and provide database access. While storing these secrets in configuration files is based on good security principles, they should not be included in the OpenEMR repo for all to see.