Security Audit

How to Do a Security Audit of Symfony, Laravel & Codeigniter Frameworks?

How to Do a Security Audit of
SYMFONY, LARAVEL AND CODEIGNITER
astra
CodeIgniter

Updated on: April 5, 2022

Jinson Varghese

8 mins read

PHP is still widely used to develop websites and open-source CMSes. However, developing large applications using PHP can turn out to be time-consuming. So to overcome that, certain frameworks like Symfony, Laravel, and Codeigniter are used.

This Blog Includes [show]

These frameworks come with certain pre-built components that can be reused to develop web applications faster and easier. However, even these frameworks are not entirely secure. A small vulnerability in any one of them can seep down to thousands of web apps developed using them. To check such bugs beforehand you can conduct a Symfony security audit, Laravel security audit or Codeigniter security audit.

Today we are going to discuss the common security issues in these frameworks and how to overcome them with a thorough security audit.

1. Symfony

The official website of Symfony describes it as,

> Symfony is a set of reusable PHP components and a PHP framework for web projects.

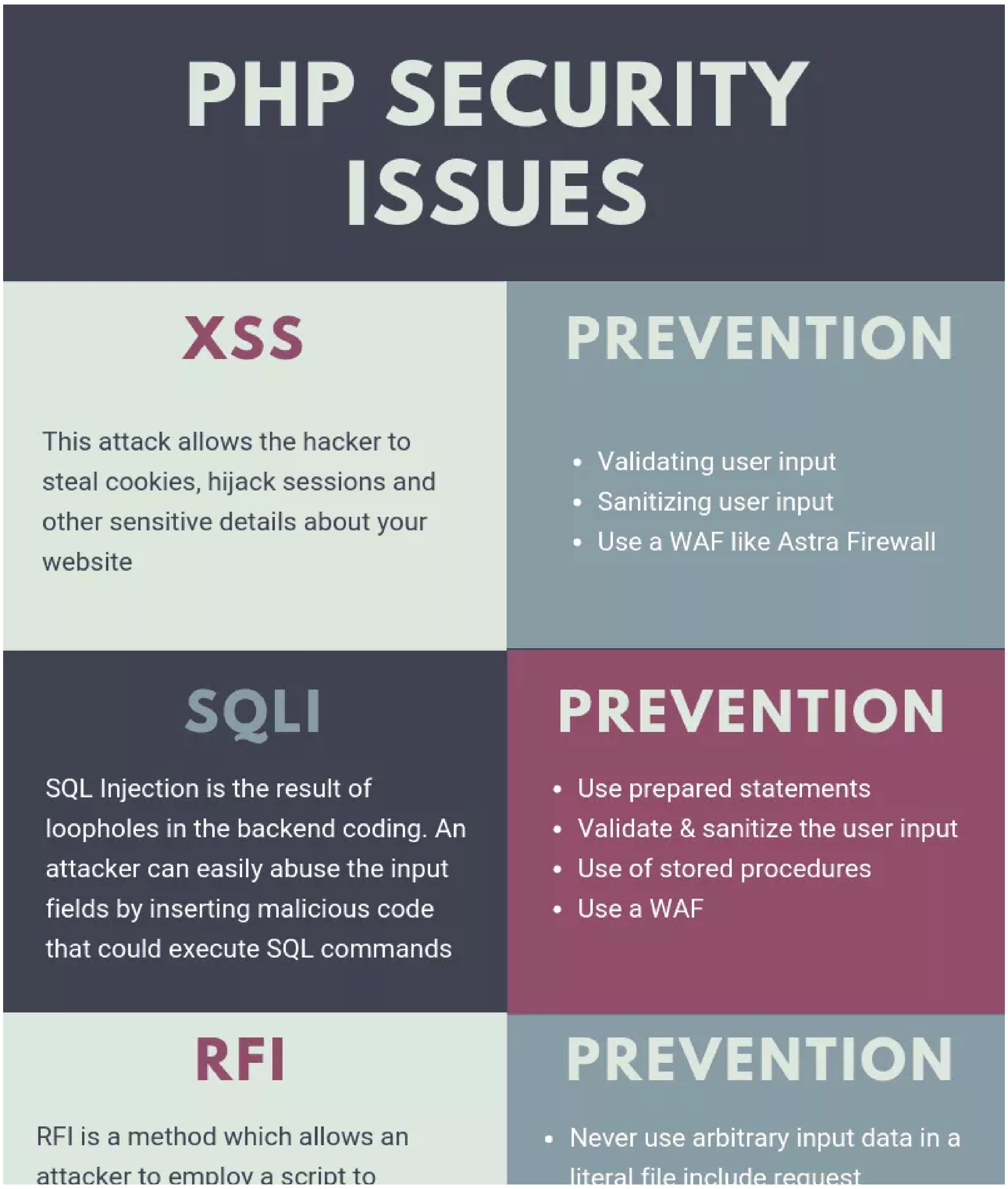
And rightly so. It contains a collection of frequently used components. For example, while building multiple web applications, you will need an authentication component of some sort. So, instead of re-building the login component every time, Symfony can create one for you on the go. Similarly, it contains other commonly used components like forms, filesystem, etc.

Need for Symfony Security Audit

Building your web app using Symfony is easy, however, it does not imply that it is free of vulnerabilities. Symfony has had its fair share of vulnerabilities in the past. Many of its components were found vulnerable to XSS, SQLi, etc attacks. For instance, **symfony/http-foundation** was found vulnerable to SQLi and XSS bugs termed as **CVE-2019-10913**.

In all these cases, the vulnerability of one component could have compromised thousands of web apps using that component.

To avoid such a scenario, it becomes very necessary to find and patch these vulnerabilities. Conducting a Symfony security audit helps you with that and more.



PHP SECURITY ISSUES

XSS

This attack allows the hacker to steal cookies, hijack sessions and other sensitive details about your website

PREVENTION

- Validating user input
- Sanitizing user input
- Use a WAF like Astra Firewall

SQLI

SQL Injection is the result of loopholes in the backend coding. An attacker can easily abuse the input fields by inserting malicious code that could execute SQL commands

PREVENTION

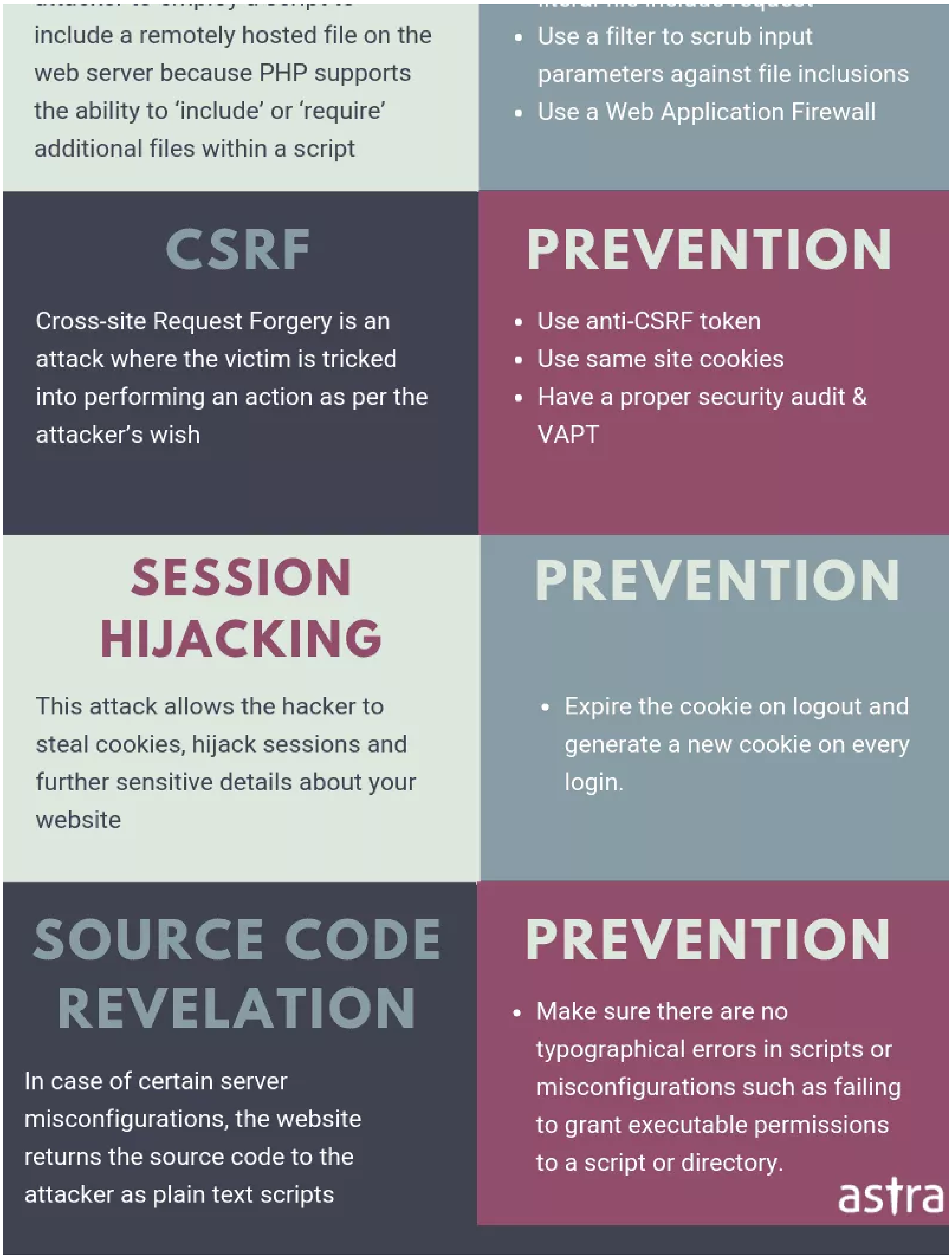
- Use prepared statements
- Validate & sanitize the user input
- Use of stored procedures
- Use a WAF

RFI

RFI is a method which allows an attacker to employ a script to

PREVENTION

- Never use arbitrary input data in a literal file include request



attacker to employ a script to include a remotely hosted file on the web server because PHP supports the ability to 'include' or 'require' additional files within a script

local file include request
- Use a filter to scrub input parameters against file inclusions
- Use a Web Application Firewall

CSRF

Cross-site Request Forgery is an attack where the victim is tricked into performing an action as per the attacker's wish

PREVENTION

- Use anti-CSRF token
- Use same site cookies
- Have a proper security audit & VAPT

SESSION HIJACKING

This attack allows the hacker to steal cookies, hijack sessions and further sensitive details about your website

PREVENTION

- Expire the cookie on logout and generate a new cookie on every login.

SOURCE CODE REVELATION

In case of certain server misconfigurations, the website returns the source code to the attacker as plain text scripts

PREVENTION

- Make sure there are no typographical errors in scripts or misconfigurations such as failing to grant executable permissions to a script or directory.

Security issues in PHP

2. Laravel

Similar to Symfony, Laravel is a framework that makes it easy to develop web applications. Laravel is very diversified and has a big web ecosystem. According to the official website of Laravel,

> We believe development must be an enjoyable and creative experience to be truly fulfilling. Laravel attempts to take the pain out of development by easing common tasks used in most web projects.

Need For Laravel Security Audit

Although the Laravel code is checked by the community for security errors, yet one or two usually slips by. For instance, the Laravel framework versions 5.5.40 and 5.6.x through 5.6.29, were found vulnerable to remote code execution. Termed as **CVE-2018-**

15133, this bug was caused due to a vulnerable **X-XSRF-TOKEN** value. What is more alarming is that there exists a Metasploit module to exploit the same!

This makes conducting a Laravel security audit a must. It can help in checking your web app for any such vulnerabilities and save you from any misfortune whatsoever. A vulnerability assessment also helps in patching them before the web app goes live.

3. Codeigniter

In the series of PHP frameworks mentioned above, coming next is Codeigniter. Codeigniter is another powerful and lightweight framework used to build web apps. It also consists of clear documentation that helps anyone learn it efficiently. Another big advantage of Codeigniter is that it offers a simple routing method.

The official website of Codeigniter defines it as,

> CodeIgniter is a powerful PHP framework with a very small footprint, built for developers who need a simple and elegant toolkit to create full-featured web applications.

Need for Codeigniter Security Audit

Using a framework to develop PHP web apps does not always guarantee security. For instance, CodeIgniter prior to 3.1.3 was found vulnerable to a remote code execution bug. This was termed as **CVE-2016-10131** and was caused due to vulnerable **system/libraries/Email.php**.

This meant that all the web apps which had used the Email.php of Codeigniter before 3.1.3 were also vulnerable to RCE. To catch such bugs before hackers exploit them, performing a Codeigniter security audit is necessary. Indulging in a periodic security audit ensures the safety of your web app in the long run.

How to perform a security audit of Symfony, Laravel or Codeigniter?

Since all the above-mentioned frameworks use PHP, a set of common tools can be used for the security audit of web apps developed using these frameworks. These tools can be installed manually too but it is advisable to use Kali Linux for this purpose for Kali Linux comes preloaded with most of these tools.

If you are a window user and wish to use it using a virtual box, you can set it up like this.

Once the setup is done, you are ready for the security audit part so let's dive in!

1. PhpStan

This tool is widely used to security audit the static code of the PHP web app generated by the above-mentioned frameworks. This tool is not included in the official Kali bundle so you will have to download and install it manually. Once installed, you are ready to use PhpStan. Now suppose the files of your web app are in the tests and src folders. Then, open the terminal and run the following command:

```
vendor/bin/phpstan analyse src tests
```

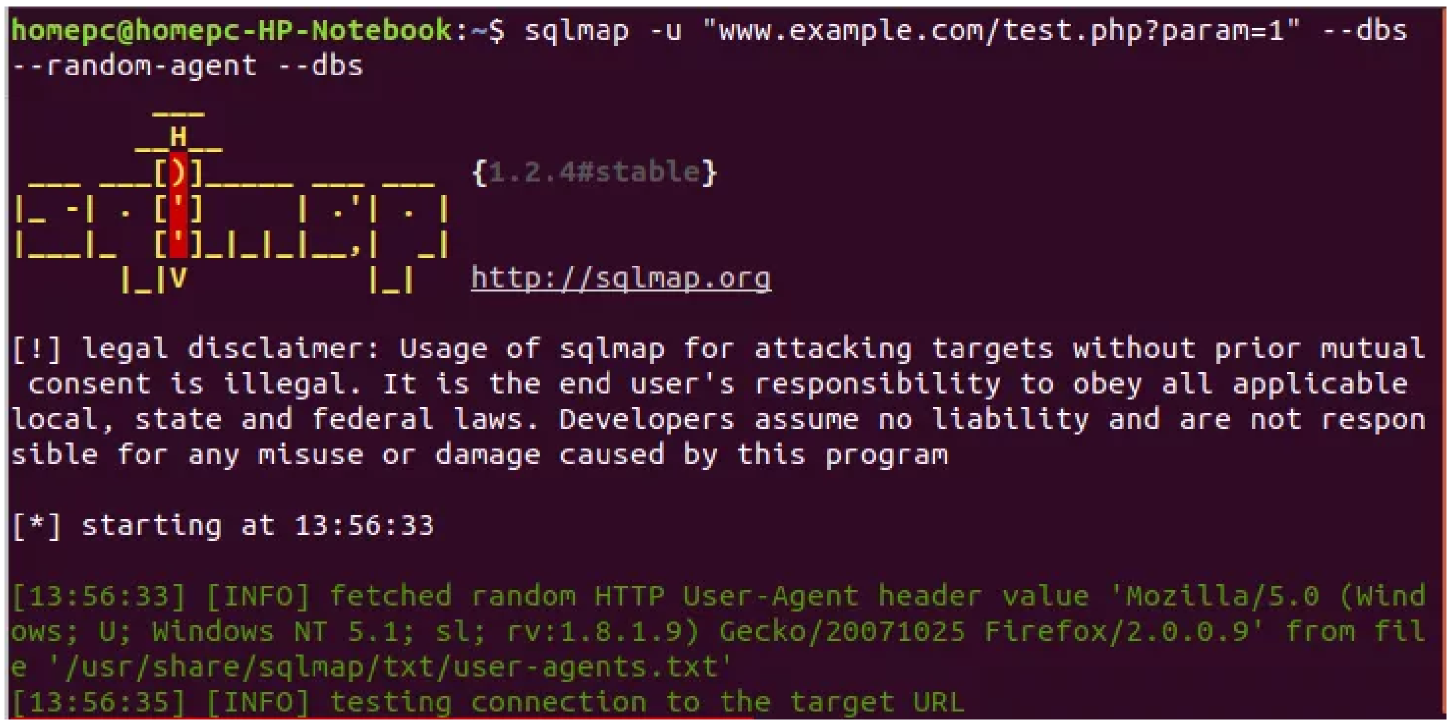
Moreover, PhpStan has unofficial extensions specifically for Laravel known as Larastan and one for Symfony too. So you might want these based on specific frameworks.

2. Sqlmap

The SQL injection is one of the most common vulnerabilities found in the web apps during a Symfony, Laravel or Codeigniter security audit. This bug can be hunted by using Sqlmap. You can either test your web app for SQLi bugs live on the internet or you can test them on your local server. For example, the page you wish to test for SQLi is "**test.php**" and parameter is "**param**" then, open the terminal in Kali and type:

```
sqlmap -u "www.example.com/test.php?param=1" --dbs --random-agent --dbs
```

This command will try to check **test.php** for SQLi vulnerability. If present, Sqlmap will try to enumerate database names too. For more detailed usage options, refer to the official documentation.



```
homepc@homepc-HP-Notebook:~$ sqlmap -u "www.example.com/test.php?param=1" --dbs
--random-agent --dbs
        ___
       __H__
 ___ ___[)]_____ ___ ___  {1.2.4#stable}
|_ -| . [']     | .'| . |
|___|_  [']_|_|_|__,|  _|
      |_|V          |_|   http://sqlmap.org

[!] legal disclaimer: Usage of sqlmap for attacking targets without prior mutual
 consent is illegal. It is the end user's responsibility to obey all applicable
local, state and federal laws. Developers assume no liability and are not respon
sible for any misuse or damage caused by this program

[*] starting at 13:56:33

[13:56:33] [INFO] fetched random HTTP User-Agent header value 'Mozilla/5.0 (Wind
ows; U; Windows NT 5.1; sl; rv:1.8.1.9) Gecko/20071025 Firefox/2.0.0.9' from fil
e '/usr/share/sqlmap/txt/user-agents.txt'
[13:56:35] [INFO] testing connection to the target URL
```

3. Xsser

Another most common bug found during a Laravel, Codeigniter or Symfony security audit is an XSS vulnerability. These can be hunted down using Xsser; a tool to discover as well as exploit XSS bugs by bypassing security filters.

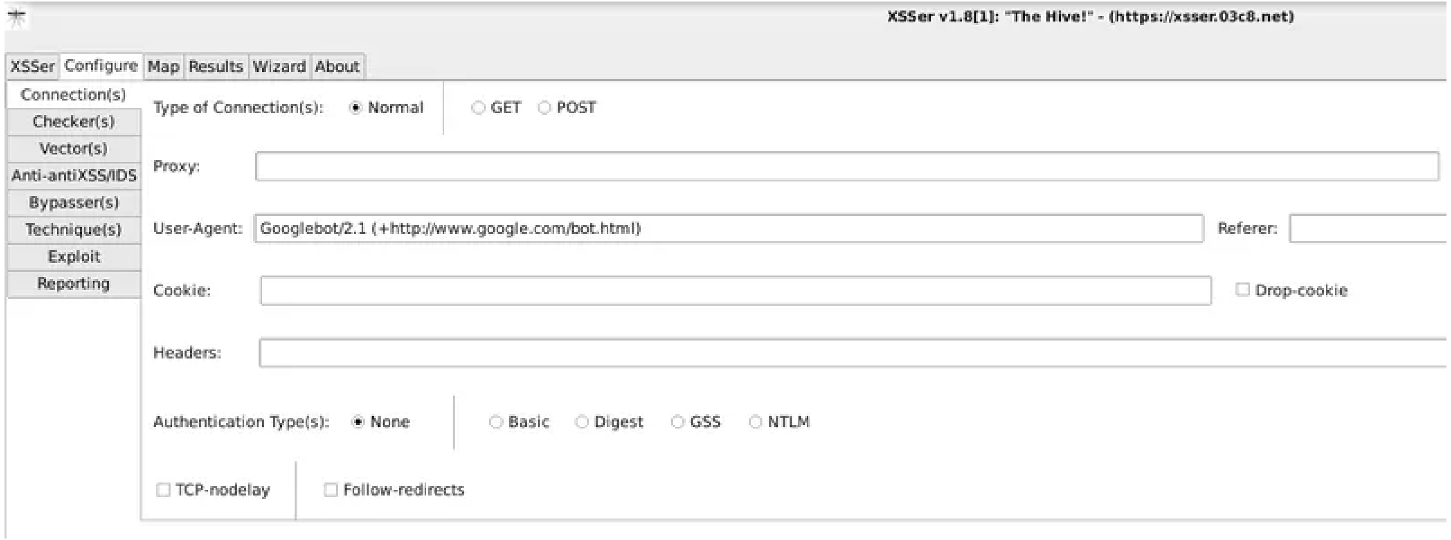
Beginners may find it easy to use this tool using the graphical interface. So, open the terminal in Kali and type:

```
xsser --gtk
```

In case this command does not work or you have not set the environment variables then try:

```
python3 xsser --gtk
```

This will open a graphical interface like the one in the image below. Just set the necessary options and start finding XSS bugs. For more details on the options, refer to the documentation of this tool.



XSSer v1.8[1]: "The Hive!" - (https://xsser.03c8.net)

Type of Connection(s): ● Normal ○ GET ○ POST

Proxy:

User-Agent: Googlebot/2.1 (+http://www.google.com/bot.html) Referer:

Cookie: □ Drop-cookie

Headers:

Authentication Type(s): ● None ○ Basic ○ Digest ○ GSS ○ NTLM

□ TCP-nodelay □ Follow-redirects

4. Fimap

File inclusion vulnerabilities are also commonly found in most web apps along with XSS and SQLi bugs. To scan your web app developed using Laravel, Symfony or Codeigniter for file inclusion bugs, Fimap can be used.

For example, if you wish to scan the "**index.php**" page for file inclusion vulnerabilities, open the terminal in Kali and run the following command:

```
fimap -u "http://www.example.com/index.php"
```

For more info on the usage, in the terminal type:

```
fimap -h
```

Security Audit Service for Symfony, Laravel, Codeigniter

This article is just an introduction to Symfony, Codeigniter, and Laravel security audit. it barely scratches the surface and covers only the basics. There is a lot that needs to be checked to ensure the security of web apps developed using these frameworks.

This is why to ensure maximum security it is recommended to go for a professional security audit. Astra contains a very flexible security and penetration testing plan. It doesn't matter if you own a small blog or run an online store, Astra has something for everyone at very affordable prices. Its comprehensive security audit covers major issues like:

- Configuration and Deployment Mis-configuration.
- PHP Core, Plugins & Theme Specific Vulnerabilities.
- Broken or Improper Authentication.
- Identifying Technical & Business Logic Vulnerabilities.
- 1250+ Active Security Tests.

Still, have some doubts? Leave a comment below or drop us a message using the chat widget.

Was this post helpful?

Yes	3
No	1

Jinson Varghese

Jinson Varghese Behanan is an Information Security Analyst at Astra. Passionate about Cybersecurity from a young age, Jinson completed his Bachelor's degree in Computer Security from Northumbria University. When he isn't glued to a computer screen, he spends his time reading InfoSec materials, playing basketball, learning French and traveling. You can follow him on Medium or visit his Website for more stories about the various Security Audits he does and the crazy vulnerabilities he finds.

Be the First to Comment!

B I U S [+]

This site uses Akismet to reduce spam. Learn how your comment data is processed.

0 COMMENTS

Related Articles

Security Audit

Risk Assessment vs Vulnerability Assessment: A Detailed Discussion

Security Audit

A Complete Guide on Vulnerability Assessment Methodology

Security Audit

A Quick Guide to PCI Penetration Testing

Psst! Hi there. We're Astra.

We make security simple and hassle-free for thousands of websites and businesses worldwide.

Our suite of security products include a vulnerability scanner, firewall, malware scanner and pentests to protect your site from the evil forces on the internet, even when you sleep.

Get a Pentest

Protect your website

We make security simple and hassle-free for thousands of websites & businesses worldwide.

+ Pentest

+ Website Protection

+ Company

+ Resources

Secured by astra
getastra.com

See our glowing reviews on

Trustpilot Capterra

Made with ❤️ in

Privacy Policy | **Terms of Service** | **Report a vulnerability**

Secured by astra

getastra.com