2023/4/26

Project Report: Secure Grouping System

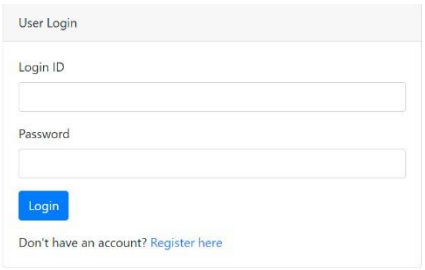
CHEUK, cyrus [Student]
[公司名稱]

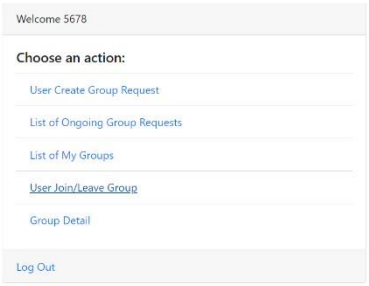
Contents

I. Introduction

1.1Web UI Design

The web application project is a group management system that allows users to create groups, manage group membership, and share content within groups. The application has a responsive web UI that is designed using PHP, HTML, CSS, and Bootstrap. The UI is intuitive and user-friendly, with features such as easy navigation, clear labeling, and search functionality. It allows users to create and join groups. Upon successful login, users are redirected to the protected area of the homepage, which includes various features and functions. All pages in the protected area require authentication and authorization to access. The homepage of the website displays a list of all available groups, and users can click on a group to view more details about it. Users can create their own groups by filling out a form with a title, description, preferred size, and other information. Once a group is created, other users can request to join the group. The web UI also includes a login system to ensure that only authorized users can create and join groups.





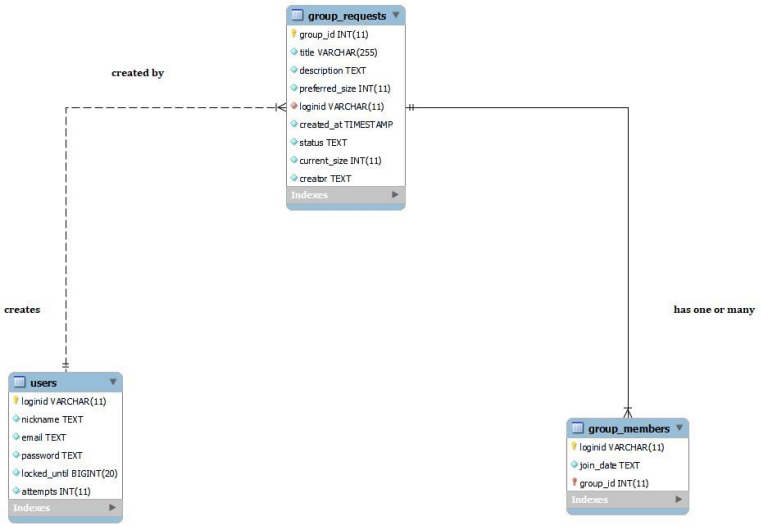
1.2Database Design:

The database is built on MySQL, and the application uses PHP to interact with the database. It was designed to store information about users and groups. The database design includes tables for storing user information, group information, and content shared within groups. The 'users' table contains user login information, including their login ID, nickname, email, password, and other data related to login attempts.In the users table, there are two additional columns named locked_until and attempts. These columns are used to track failed login attempts by a user and lock the account if there are too many failed attempts.

The locked_until column stores a Unix timestamp indicating the time until which the account is locked. If the value is 0, it means the account is not locked. If the value is greater than or equal to 5, it means the account is locked until that Unix timestamp.

The attempts column stores the number of failed login attempts made by the user. If the user enters incorrect login credentials, this value is incremented. If the user enters correct login credentials after the timestamp , this value is reset to 0.

When a user exceeds the maximum number of login attempts, their account is locked until the current time plus the lockout duration, which is stored in the locked_until column. The 'group_requests' table contains information about the groups that have been created, including the group ID, title, description, preferred size, login ID of the creator, and other data related to group creation. The 'group_members' table contains information about the users who have joined a particular group, including their login ID, join date, and the group ID. Additionally, the system includes a mechanism to limit the number of login attempts and lock accounts temporarily after a certain number of failed attempts The database connection is established using PDO, and prepared statements are used to prevent SQL injection attacks.



1.3Network Design:

The website is hosted on a local XAMPP server, which allows for easy testing and development. The network design is based on a client-server architecture, where the web application is hosted on a server and accessed by users via web browsers. The server is connected to a local network, which allows other users on the network to access the website. Moreover, the application uses HTTPS with SSL 1.3 encryption to ensure secure communication between the client and server. A pfsense used as firewall port forwarding and related extension snort is also set up to restrict access to the application's ports and block unauthorized requests. Additionally, HTTP headers are configured to enforce HTTPS connections and protect against certain types of attacks such as Cross-Site Scripting (XSS) and Clickjacking. This helps to ensure that the network is secure, and the application is protected from potential threats. Also, using Pfsense extensions to defend ddos.

2. Tools and Techniques and Security evaluation

Tools: The system is built using PHP, pfSense as firewall and port forwarding, VMWare Workstation Pro for hosting and various security measures have been implemented to ensure the system is secure against common attacks.

Security evaluation :

1. Authentication and Authorization:

- Ensure that only authorized users have access to sensitive information and functionalities.
- Implement strong password policies, at least 8 characters, and account lockout mechanisms to prevent unauthorized access.

2. Use hash encryption to protect sensitive data, such as passwords.

3. Secure Communication:

- Implement HTTPS/TLS to encrypt all data transmitted between the server and clients.
- Set HTTP headers, such as X-Content-Type-Options, X-Frame-Options, and X-XSS-Protection, to protect against common web attacks.
- Use secure cookies, such as HttpOnly, Secure, and SameSite, to prevent cookie-based attacks.

3. Server Security:

- Implement firewalls, intrusion detection and prevention systems, o detect and prevent attacks with using extensions.
- Disable or remove unnecessary server services to reduce the attack surface.

4. Database Security:

- Use secure coding practices to prevent SQL injection and other database attacks.
- Implement proper access controls and permissions to ensure that only authorized users can access, modify, or delete data.
- Encrypt sensitive data at rest and in transit.

5. Regular Security Audits:

- Conduct regular security audits to identify and address vulnerabilities and weaknesses in your web system.

- Use security testing tools, such as ZAP to perform vulnerability scanners, penetration testing, and code analysis tools, to identify and mitigate security risks.

Some of the tools and techniques used include:

1.Password hashing using bcrypt algorithm

2. session_set_cookie_params and related configuaration for a safe session cookie and normal cookie.

```
e.g. session_set_cookie_params([
    'lifetime' => 0,
    'path' => '/',
    'domain' => '',
    'secure' => true,
    'httponly' => true,
    'samesite' => 'Strict'
]);
```

3. Password at least 8 characters

4. Anti-CSRF Token using bin2hex(random_bytes(32)); and hidden filed contain token when submitted.

If token not hash_equals, then kill the program. Regenerate automatically. Prevent CSRF

5. used filter_var(($_POST['loginid']), FILTER_SANITIZE_STRING); and htmlspecialchars for avoid XSS.

6.Some input value by user restricted to only value

7.Prepared statements to prevent SQL injection attacks

8.Limiting login attempts and locking accounts temporarily after a certain number of failed attempts

9.Use of HTTPS to encrypt all data transmitted between the client and server

	loginid	nickname	email	password	locked_until	attempts
Edit Copy Delete	1234	1234@gmail.com	1234@gmail.com	$2y$10$RSVEr/dW9qW58JxaNAIFZuIOYnQ1prrsScuXB9h5wFb...	1682433850	5

10.Implementation of various HTTP response headers to prevent common web-based attacks

11. Applied Content-Security-Policy

12. Applied **Header set Content-Security-Policy at htacess and php**

13. Header set X-Content-Type-Options

14.Header always set Strict-Transport-Security

15. Hide apache_server_info

16.The following tools and techniques were also used to implement security measures to prevent DDoS attacks for website:

- Extension Mod_evasive: This Apache module was used to detect and block HTTP-based DDoS attacks. It monitors incoming requests and blocks IP addresses that exceed a specified threshold of requests.
- Rate-limiting: The mod_evasive module was configured to rate-limit requests from individual IP addresses or ranges to prevent them from overwhelming the server.
- Firewall rules: Firewall rules were implemented to block incoming traffic from known malicious IP addresses and ranges.
- Using snort extension for Pfsense to provides real-time network traffic analysis and data packet logging to avoid ddos.
- PfblockerNG to block network traffic from a specific IP to prevent ddos

The system has been evaluated using various security testing tools, including vulnerability scanners and penetration testing tools. No major vulnerabilities or issues were identified during the testing phase, and the system is considered secure against common attacks. Also, have applied ZAP to detect the vulnerabilities of the website of grouping system. The security measures implemented were able to detect and block DDoS attacks on the web server. The mod_evasive module was able to successfully rate-limit incoming requests from individual IP addresses and ranges, which helped to prevent the server from becoming overwhelmed. The mod_security module helped to detect and block malicious traffic at the application layer, which helped to mitigate some types of DDoS attacks.

3. Difficulties Encountered

During the implementation phase, various technical and non-technical difficulties were encountered. Some of the technical difficulties included: Setting up and configuring HTTPS on the server. Also, I need to implement various HTTP response headers to prevent common web-based attacks. Moreover, some of the difficulties encountered during the project included issues with compiling and installing the Apache modules on a Windows operating system. In addition, there

were some challenges in configuring the mod_evasive module to properly rate-limit incoming requests. These challenges were solved by adjusting the configuration settings and testing the module to ensure that it was effectively blocking requests.

Non-technical difficulties included:

1.Ensuring the system was user-friendly and easy to use

2.Balancing security with usability

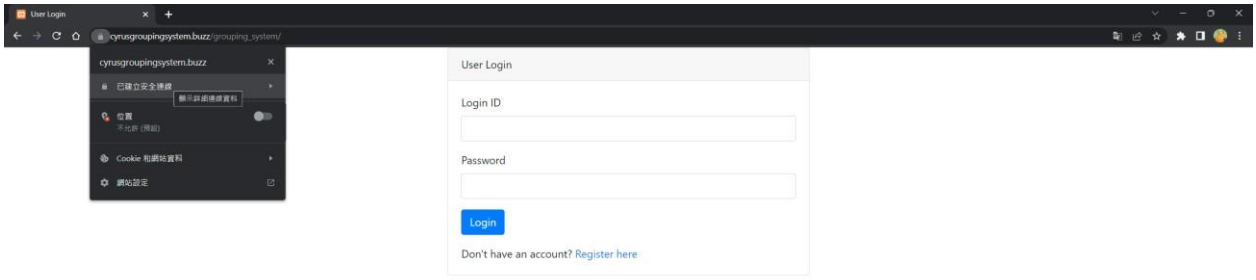
Therefore, I need to keep doing Website testing to improve the user experience.

4. Conclusion

In conclusion, the grouping system has been built using various tools and techniques to ensure the system is secure against common attacks. The system has been thoroughly tested and evaluated, and no major vulnerabilities or issues were identified. With ongoing maintenance and updates, the grouping system will continue to provide a secure and reliable application for users.

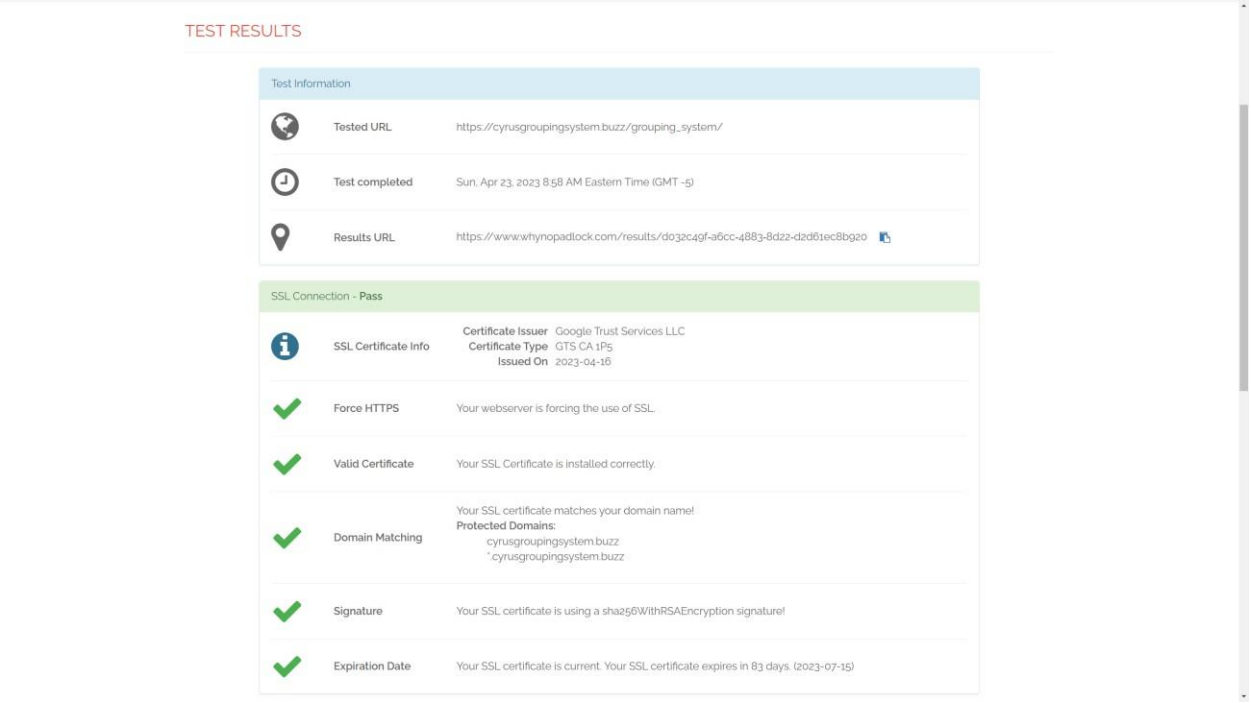
5. Security Demo

1.1 HTTPS (demo by screenshots)



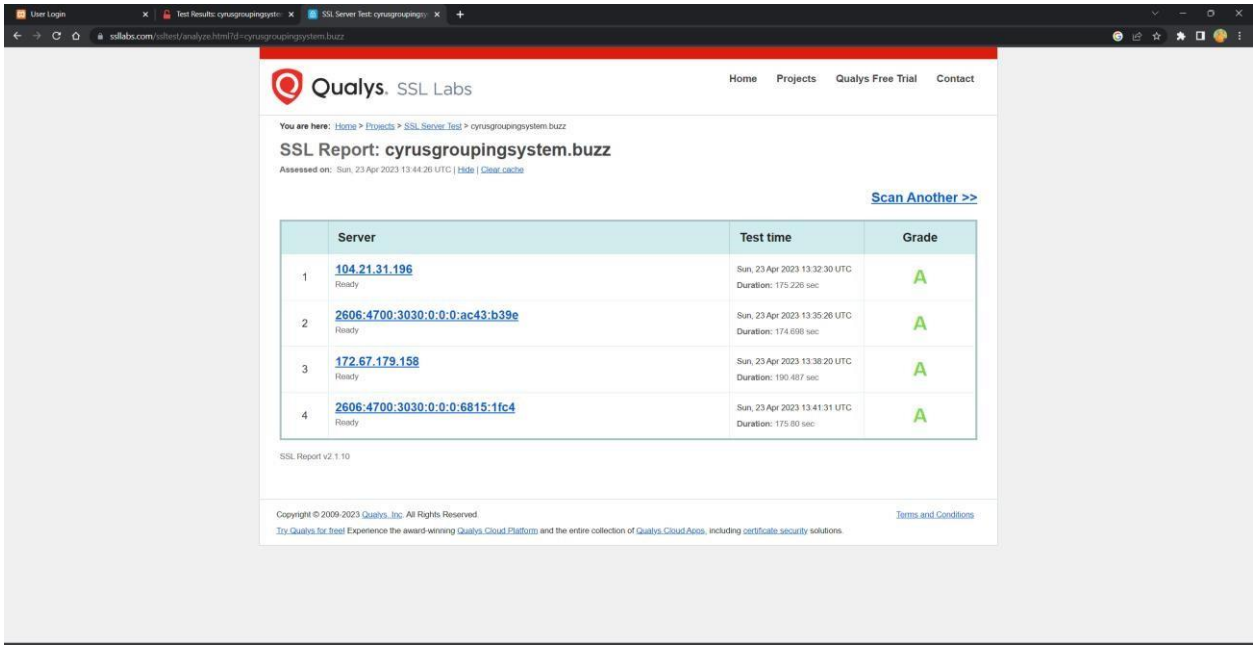
1. Check whether your web contents are secured by https through:

https://www.whynopadlock.com/index.html



2. Check if your certificate information is correct through: https://www.ssllabs.com/ssltest

Requirement: get an “D” or above grade (“T” score means your certificate is not trusted).

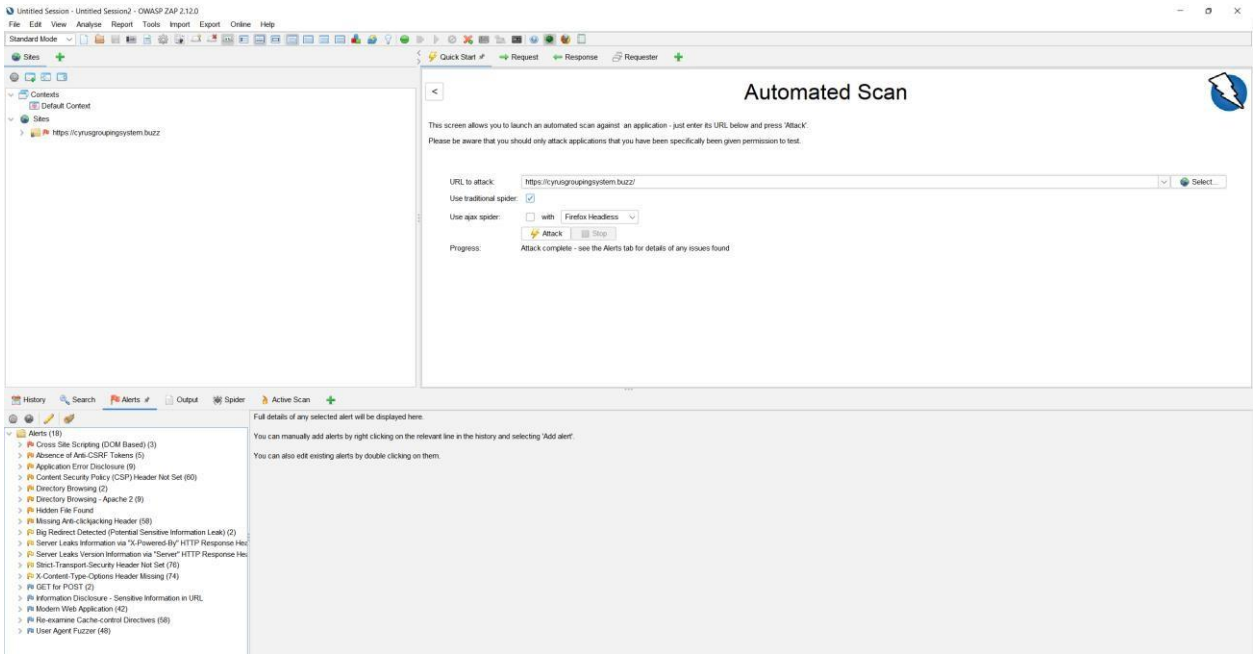


1.2 Web Vulnerabilities Check

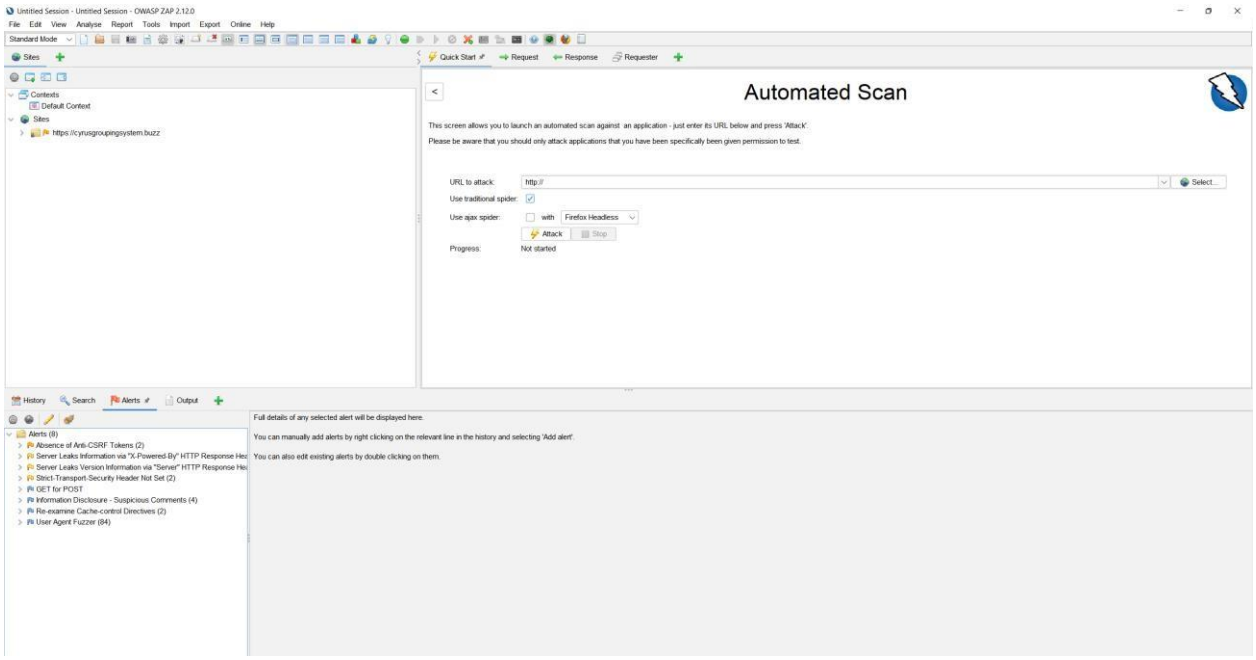
Show your latest result of Web Application Tests by Zed Attack Proxy (available as a stress test tool in Kali Linux).

You need to demonstrate the results before and after the code audit and revision. To show your efforts, you should solve all “major” vulnerabilities except for those uncontrollable ones (for example some vulnerabilities appear simply because the WAMP/LAMP distribution you use does not use the latest version of PHP). You might be asked to show samples of code revision.

Before audition:



After audition:



Fixed XSS(DOM BASED)

Fixed Application Error Disclosure

Fixed CSP header Not Set

Fixed Directory Browsing

Fixed Directory Browsing-Apache2

Fixed Hidden File Found

Fixed Missing Anti-Clickjacking Header

Fixed Big Redirect Detected (potential Sensitive Information Leak)

Fixed X-Content-type-Options Header Missing

Also have applied Anti-CSRF Token :

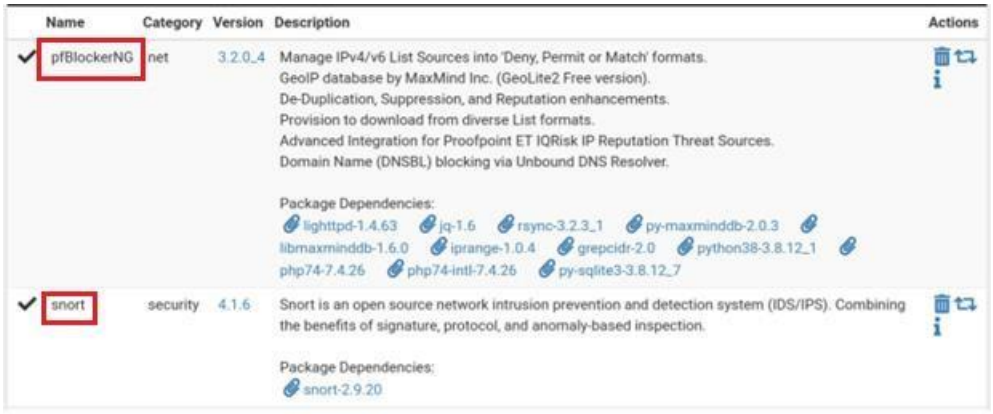
```php
1 reference
function generateCSRFToken() {
    $token = bin2hex(random_bytes(32));
    $_SESSION['csrf_token'] = $token;
    return $token;
}
```

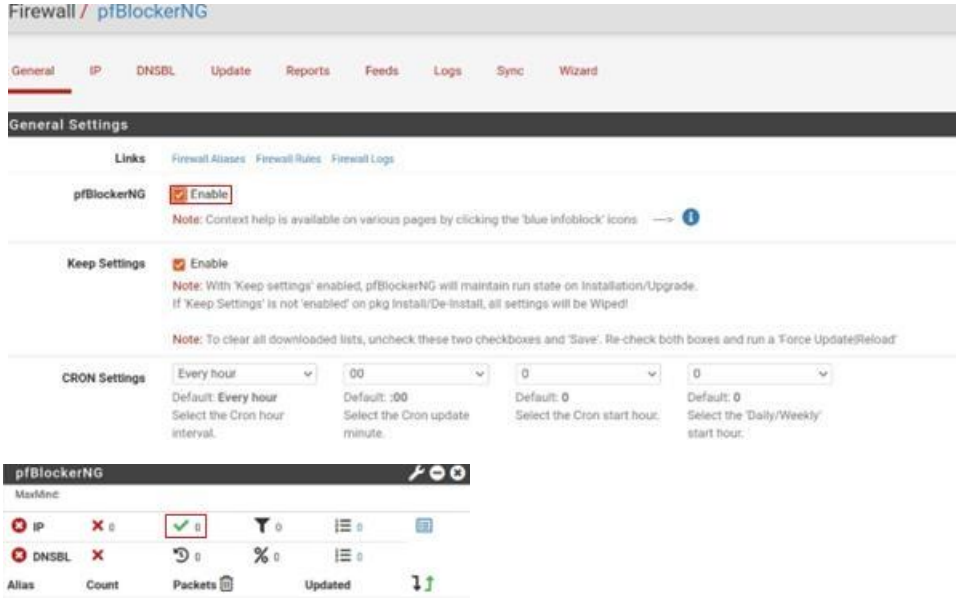
```php
// Compares two strings using the same time whether they're equal or not. because sometime there is some time difference if the token is close to the correct one
if (!isset($_POST['csrf_token']) && hash_equals($_SESSION['csrf_token'],$_POST['csrf_token'])) {// mitigate timing attacks
    //measure the time difference between two comparisons and deduce that the second string is more correct (since the second comparison took more time).
    //hash_equals eliminates this type of attack by always comparing all characters of both strings, not matter if they match or not. So more and less correct strings will take the same time.
    die("CSRF token verification failed");
}
```

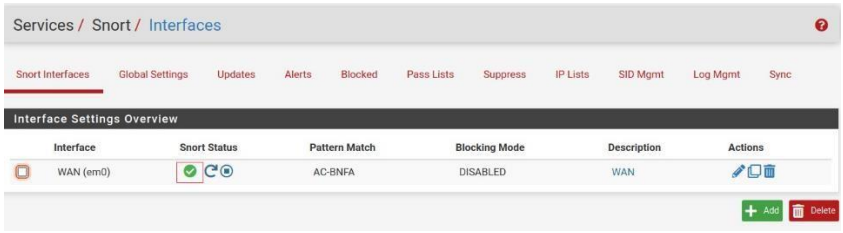
```php
    <form method="POST" action="index.php">

    <!-- adding a hidden field to your form that contains a value that only you and your user know.
    This ensures that the user - not some other entity - is submitting the given data. -->
    <input type="hidden" name="csrf_token" value="<?php echo generateCSRFToken(); ?>">
```

1.3 Firewall Configuration

Show the web admin interface of the firewall. Answer questions regarding security hardening of your application on Firewall side.



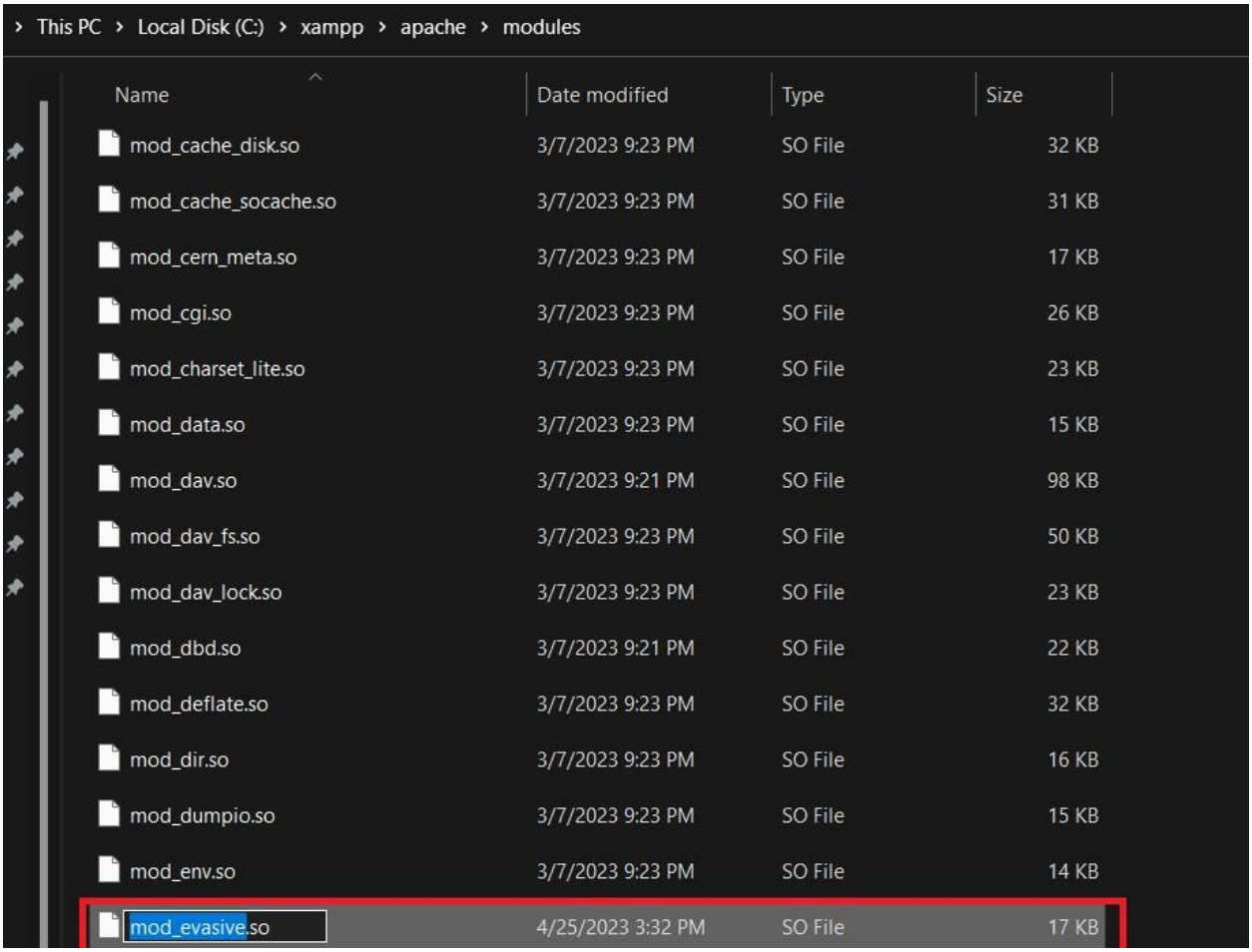




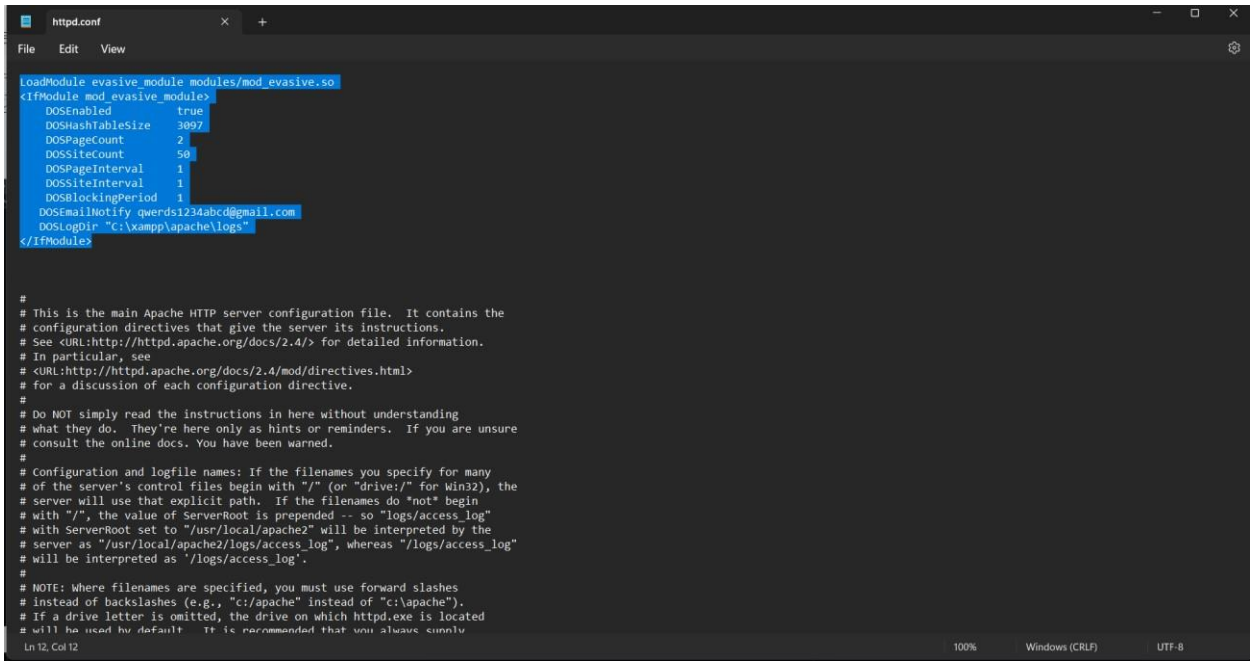
1.4 Server Hardening

Install mod_evasive2 (if you use Apache) or similar plugins/extensions (if you use other web servers such as Nginx) for anti-DDOS and web application security.

Note: This part (Section 2.4) will be skipped during the demo due to limited time, but you still need to complete it and show it in your final report.




Select Command Prompt
Microsoft Windows [Version 10.0.22000.1817]
(c) Microsoft Corporation. All rights reserved.
C:\Users\User>httpd -M
Loaded Modules:
core_module (static)
win32_module (static)
mpm_winnt_module (static)
http_module (static)
so_module (static)
evasive_module (shared)
access_compat_module (shared)
actions_module (shared)
alias_module (shared)
allowmethods_module (shared)
asis_module (shared)
auth_basic_module (shared)
authn_core_module (shared)
authn_file_module (shared)
authz_core_module (shared)
authz_groupfile_module (shared)
authz_host_module (shared)


httpd.conf
File
Edit
View
LoadModule evasive_module modules/mod_evasive.so
<IfModule mod_evasive_module>
DOSEnabled true
DOSHashTableSize 3097
DOSPageCount 2
DOSSiteCount 50
DOSPageInterval 1
DOSSiteInterval 1
DOSBlockingPeriod 1
DOSEmailNotify qwerds1234abcd@gmail.com
DOSLogDir "C:\xampp\apache\logs"
</IfModule>
#
This is the main Apache HTTP server configuration file. It contains the
configuration directives that give the server its instructions.
See <URL:http://httpd.apache.org/docs/2.4/> for detailed information.
In particular, see
<URL:http://httpd.apache.org/docs/2.4/mod/directives.html>
for a discussion of each configuration directive.
#
Do NOT simply read the instructions in here without understanding
what they do. They're here only as hints or reminders. If you are unsure
consult the online docs. You have been warned.
#
Configuration and logfile names: If the filenames you specify for many
of the server's control files begin with "/" (or "drive:/" for Win32), the
server will use that explicit path. If the filenames do *not* begin
with "/", the value of ServerRoot is prepended -- so "logs/access_log"
with ServerRoot set to "/usr/local/apache2" will be interpreted by the
server as "/usr/local/apache2/logs/access_log", whereas "/logs/access_log"
will be interpreted as '/logs/access_log'.
#
NOTE: Where filenames are specified, you must use forward slashes
instead of backslashes (e.g., "c:/apache" instead of "c:\apache").
If a drive letter is omitted, the drive on which httpd.exe is located
Ln 12, Col 12
100%
Windows (CRLF)
UTF-8

6. References

Where developers learn, share, & build careers (no date) *Stack Overflow*. Available at: https://stackoverflow.com/ (Accessed: April 26, 2023).

Documentation (no date) *php*. Available at: https://www.php.net/docs.php (Accessed: April 10, 2023).

iThome (no date) *It* 邦幫忙*::*一起幫忙解決難題，拯救 *it* 人的一天, *iT* 邦幫忙*::*一起幫忙解決難題，拯救 *IT* 人的一天. Available at: https://ithelp.ithome.com.tw/ (Accessed: April 26, 2023).

Where good ideas find you. (no date) *Medium*. Available at: https://medium.com/ (Accessed: April 26, 2023).