LKS SMK

Attack/Defense

NAMA TIM	SMKN 22 JAKARTA
ANGGOTA	Muhammad Arsyah Al Bassam
	Sholahuddin Muhammad Athar

Attack

Step :

Kami melihat ada port ssh yang belum diubah port number nya. Ketika kami mencoba login ke ssh menggunakan user root dengan password default yang diberikan oleh juri ternyata tim tersebut belum melakukan patching service ssh.

Kami tinggal login user root dan membuka flag nya dengan command *cat root.txt*, kemudian kami pergi ke home direktori user adrian dan terdapat user.txt yang berisi flag.
Dan kami berhasil mendapatkan kedua flag pada server tersebut.

*note : tidak ada screenshots, karena ketika kami ingin membuat write up, server tersebut sudah melakukan hardening pada service ssh nya.

Defense

Step :

Kami melakukan hardening di beberapa service seperti ssh dan web server.

Service SSH :
Pertama kami mengubah port, dan merubah beberapa konfigurasi pada /etc/ssh/sshd.conf seperti gambar di bawah ini:

```
Port 54545
PermitRootLogin yes
#StrictModes yes
MaxAuthTries 4
PubkeyAuthentication yes
HostbasedAuthentication no
# Change to yes if you don'
# HostbasedAuthentication
#IgnoreUserKnownHosts no
# Don't read the user's ~/.
IgnoreRhosts yes
PasswordAuthentication yes
PermitEmptyPasswords no
DebianBanner no
Ciphers aes128-ctr,aes192-ctr,aes256-ctr
Protocol 2

Match User root
       AuthenticationMethods password

Match User adrian
       AuthenticationMethods publickey,password
```

Web-Server :
Kami menambahkan beberapa modul yaitu modsecurity dan mod evasive lalu mengkonfigurasinya seperti gambar di bawah ini:

```
#
SecRuleEngine DetectionOnly
```

Disini kami biarkan SecRuleEngine sesuai default supaya website dapat diakses.

Lalu kami menambahkan mengganti konfigurasi dalam */etc/apache2/conf-available/security.conf* dan menambahkan HTTP Security Header seperti gambar dibawah :

```
#ServerTokens Minima
ServerTokens Prod
#ServerTokens Full
#ServerSignature Off
ServerSignature Off

Header always set X-Content-Type-Options: "nosniff"

#
# Setting this header will prevent other sites from embedding pages from this
# site as frames. This defends against clickjacking attacks.
# Requires mod_headers to be enabled.
#
Header always set X-Frame-Options: "sameorigin"
Header always set Content-Security-Policy "default-src 'self'; font-src *;img-src * data:; script-src
Header set Strict-Transport-Security "max-age=31536000; includeSubDomains; preload"
Header always set Referrer-Policy "strict-origin"
Header always set Permissions-Policy "geolocation=(),midi=(),sync-xhr=(),microphone=(),camera=(),magn

# vim: syntax=apache ts=4 sw=4 sts=4 sr noet
SecServerSignature waduh
```

Dan mengubah konfigurasi di */etc/apache2/apache2.conf* untuk mencegah directory listing :

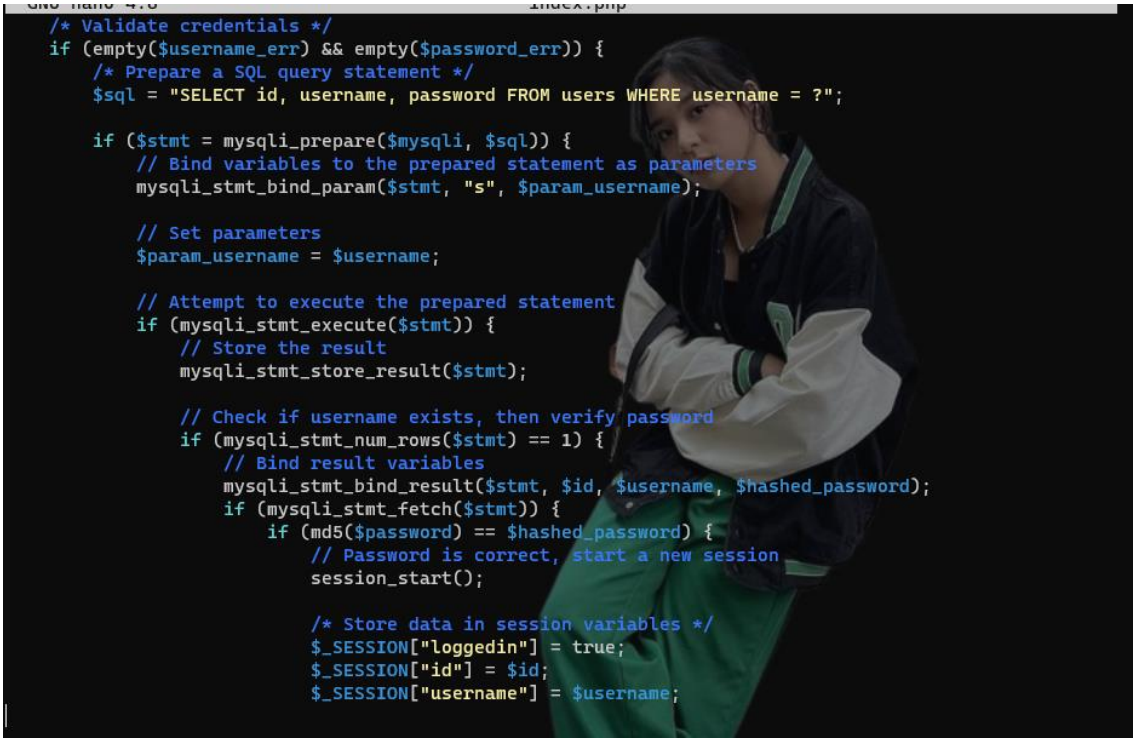
```
<Directory /var/www/>
        Options -Indexes -FollowSymLinks
        AllowOverride All
        Require all granted
</Directory>
```

Lalu kami mengubah konfigurasi *php.ini :*

```
allow_url_fopen = Off

; Whether to allow include
; http://php.net/allow-url
allow_url_include = Off
```

Setelah itu kami menyadari adanya vulnerabilty sql injection pada index.php, lalu kami merubah sedikit code pada index.php seperti gambar dibawah ini :



```php
/* Validate credentials */
if (empty($username_err) && empty($password_err)) {
    /* Prepare a SQL query statement */
    $sql = "SELECT id, username, password FROM users WHERE username = ?";

    if ($stmt = mysqli_prepare($mysqli, $sql)) {
        // Bind variables to the prepared statement as parameters
        mysqli_stmt_bind_param($stmt, "s", $param_username);

        // Set parameters
        $param_username = $username;

        // Attempt to execute the prepared statement
        if (mysqli_stmt_execute($stmt)) {
            // Store the result
            mysqli_stmt_store_result($stmt);

            // Check if username exists, then verify password
            if (mysqli_stmt_num_rows($stmt) == 1) {
                // Bind result variables
                mysqli_stmt_bind_result($stmt, $id, $username, $hashed_password);
                if (mysqli_stmt_fetch($stmt)) {
                    if (md5($password) == $hashed_password) {
                        // Password is correct, start a new session
                        session_start();

                        /* Store data in session variables */
                        $_SESSION["loggedin"] = true;
                        $_SESSION["id"] = $id;
                        $_SESSION["username"] = $username;
```