

CLOUD ARCHITECTURE AND SECURITY

CA-1

Sneha Sivaram

X23192054

URL to presentation: <https://youtu.be/M0-FFFNkyGg>

Introduction

The aim of this project is to demonstrate a detailed approach to secure a web application hosted on a cloud infrastructure. A variety of tools are used here to secure the infrastructure and pluggins for the web application. Furthermore, a thorough monitoring and testing is done on both to check for any possible vulnerabilities as well. And it is concluded with discussion of possible methods that can be carried out in future for securing the web application and the infrastructure.

The cloud infrastructure chosen here is the AWS IAAS platform, where an EC2 instance is created. And then configuring of LAMP stack (Linux, Apache, MariaDB, PHP) is done to deploy Wordpress as the web application. Following this, the security measures are taken in order to secure both the instance and the Wordpress application.

This report will cover the following objectives:

1. Launching the EC2 instance on AWS.
2. Configuring LAMP stack and Wordpress.
3. Securing and monitoring the instance.
4. Securing the Wordpress site.
5. Testing and validation.

1. Launching the EC2 instance

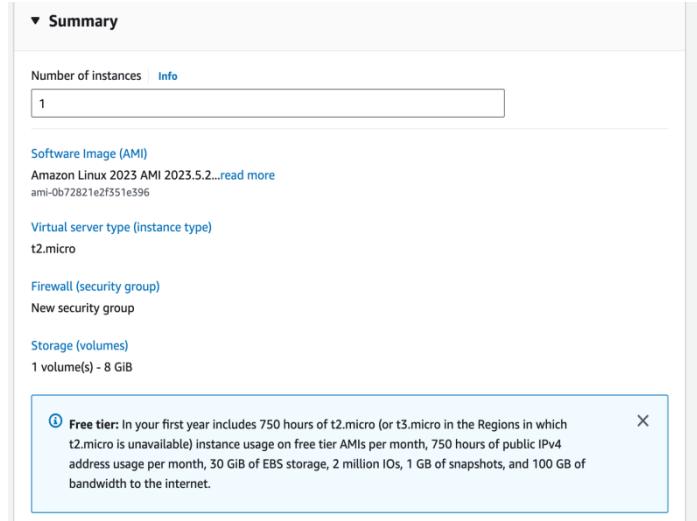
The chosen cloud platform here is AWS, where an EC2 instance is launched by setting up a proper name, choosing the OS, creating a new key pair, instance type and security groups. The ports 443, 80 were configured to accept connections from any source and the port 22 was configured to only accept from the home device's IP.

After launching the instance, a connection was established to the instance from the main device through ssh using this command:

```
ssh -i "newkey.pem" ec2-user@ec2-54-144-123-185.compute-1.amazonaws.com
```

After establishing the connection, first it is important to upgrade and update the required packages to ensure that everything is properly set up to date: `sudo dnf update -y`

The following is the details of the instance created:



2. Configuring LAMP stack and Wordpress

LAMP Stack

After establishing the connection to the instance, the initial step was to configure the LAMP stack which is also known as (Linux, Apache, MariaDB, PHP). This LAMP server can be used to host a static website or application which read and write info to the database [1].

After ensuring that all software are up to date using update command the first step was to install the latest versions of Apache webserver. PHP and MariaDB[1]:

```
sudo dnf install -y httpd wget php-fpm php-mysqli php-json php php-devel  
sudo dnf install mariadb105-server
```

The Apache webserver is then started and a command called systemctl is used to ensure that the Apache starts automatically every time the system boots [1]. After setting up Apache, the public DNS address of the instance is accessed to ensure that it working properly. Furthermore, some of the important files of Apache are stored in a root directory, so the permissions are changed from root to the ec2 user so that it allowed to manipulate any required files and so the user is added to the Apache group and the [1]:

```
sudo usermod -a -G apache ec2-user  
sudo chown -R ec2-user:apache /var/www  
sudo chmod 2775 /var/www && find /var/www -type d -exec sudo chmod 2775 {} \;  
find /var/www -type f -exec sudo chmod 0664 {} \;
```

Testing the LAMP server:

The first step is creating a PHP file in the Apache Document root.

```
echo "<?php phpinfo(); ?>" > /var/www/html/phpinfo.php
```

The phpinfo file can be accessed through the web browser using the public DNS address of the instance: <http://54.144.123.185/phpinfo.php>

The PHP file looks like this:

PHP Version 8.3.7	
System	Linux ip-172-31-92-50.ec2.internal 6.1.96-102.177.amzn2023.x86_64 #1 SMP PREEMPT_DYNAMIC Tue Jul 2 21:05:40 UTC 2024 x86_64
Build Date	May 7 2024 16:35:26
Build System	Linux
Build Provider	Amazon Linux
Compiler	gcc (GCC) 11.4.1 20230605 (Red Hat 11.4.1-2)
Architecture	x86_64
Server API	FPM/FastCGI
Virtual Directory Support	disabled
Configuration File (php.ini) Path	/etc
Loaded Configuration File	/etc/php.ini
Scan this dir for additional .ini files	/etc/php.d
Additional .ini files parsed	/etc/php.d/0-opcache.ini, /etc/php.d/20-bz2.ini, /etc/php.d/20-calendar.ini, /etc/php.d/20-ctype.ini, /etc/php.d/20-curl.ini, /etc/php.d/20-dom.ini, /etc/php.d/20-eax.ini, /etc/php.d/20-fileno.ini, /etc/php.d/20-fp.ini, /etc/php.d/20-gettext.ini, /etc/php.d/20-iconv.ini, /etc/php.d/20-mbstring.ini, /etc/php.d/20-mysqlind.ini, /etc/php.d/20-pdo.ini, /etc/php.d/20-phar.ini, /etc/php.d/20-posix.ini, /etc/php.d/20-shmop.ini, /etc/php.d/20-simplexml.ini, /etc/php.d/20-sockets.ini, /etc/php.d/20-sodium.ini, /etc/php.d/20-sqlite3.ini, /etc/php.d/20-sysvmsg.ini, /etc/php.d/20-sysvsem.ini, /etc/php.d/20-sysvshm.ini, /etc/php.d/20-tokenizer.ini, /etc/php.d/20-xml.ini, /etc/php.d/20-xmlwriter.ini, /etc/php.d/20-xsl.ini, /etc/php.d/30-mysqli.ini, /etc/php.d/30-pdo_mysql.ini, /etc/php.d/30-pdo_sqlite.ini, /etc/php.d/30-xmleader.ini
PHP API	20230831
PHP Extension	20230831
Zend Extension	420230831
Zend Extension Build	API420230831,NTS
PHP Extension Build	API20230831,NTS
Debug Build	no
Thread Safety	disabled
Zend Signal Handling	enabled
Zend Memory Manager	enabled
Zend Multibyte Support	provided by mbstring
Zend Max Execution Timers	disabled
IPv6 Support	enabled
DTrace Support	available, disabled
Registered PHP Streams	https, ftps, compress.zlib, php, file, glob, data, http, ftp, compress.bzip2, phar
Registered Stream Socket Transports	tcp, udp, unix, udg, ssl, tls, tlsv1.0, tlsv1.1, tlsv1.2, tlsv1.3
Registered Stream Filters	zlib.* , string.rot13, string.toupper, string.tolower, convert.* , consumed, dechunk, bzip2.* , convert.iconv.*

For safety reasons the PHP file is deleted in order to protect the sensitive data from any possible unauthorized access.

Securing the Database server (MariaDB)

The server is initially started and a secure installation is done to change the root password and remove anonymous accounts, disable root login and removing the test database [1].

The LAMP stack is now configured and the next step is to host the wordpress site.

Wordpress

After configuring the LAMP server the next step is to host the Wordpress site. Initially several important packages and downloaded and installed [2]:

```
dnf install wget php-mysqlnd httpd php-fpm php-mysqli mariadb105-server php-json php
php-devel -y
```

Then the latest wordpress is installed using the wget command and the installation package is unzipped to a folder:

```
wget https://wordpress.org/latest.tar.gz
tar -xzf latest.tar.gz
```

It is important to have a database for the wordpress site to store data like blog posts, users, comments etc. So, a user is created and a database is assigned and authorized to the user in order to read and save the data [2].

Firstly the mariadb is started and logged into as root with the password set during the lampstack configuration:

```
sudo systemctl start mariadb httpd
```

```
mysql -u root -p
```

A user is then created along with a password and later on a database is created and assigned to the user after granting the privileges:

```
CREATE USER 'clouduser'@'localhost' IDENTIFIED BY 'cloud123';
CREATE DATABASE `wordpress-db`;
GRANT ALL PRIVILEGES ON `wordpress-db`.* TO "clouduser"@"localhost";
FLUSH PRIVILEGES;
Exit;
```

The next step is to copy a sample configuration file called wp-config-sample.php which is present in the wordpress installation folder to edit and configure according to requirements [2].

```
cp wordpress/wp-config-sample.php wordpress/wp-config.php
nano wordpress/wp-config.php
```

In the .php file, the the database, username and password are replaced with actual ones that were created in the previous steps. And for more safety the section called Authentication Unique Key and salts values are replaced with actual generated values to ensure the safety of website [2].

```
GNU nano 5.8                               wordpress/wp-config.php                                         Modified
<?php
/*
 * The base configuration for WordPress
 *
 * The wp-config.php creation script uses this file during the installation.
 * You don't have to use the website, you can copy this file to "wp-config.php"
 * and fill in the values.
 *
 * This file contains the following configurations:
 *
 * Database settings
 * Secret keys
 * Database table prefix
 * ABS_PATH
 *
 * @link https://developer.wordpress.org/advanced-administration/wordpress/wp-config/
 *
 * @package WordPress
 */

// ** Database settings - You can get this info from your web host ** //
/** The name of the database for WordPress */
define( 'DB_NAME', 'wordpress-db' );

/** Database username */
define( 'DB_USER', 'clouduser' );

/** Database password */
define( 'DB_PASSWORD', 'cloud123' );

/** Database hostname */
define( 'DB_HOST', 'localhost' );

/** Database charset to use in creating database tables. */
define( 'DB_CHARSET', 'utf8' );

/** The database collate type. Don't change this if in doubt. */
define( 'DB_COLLATE', '' );

/**#@+
 * Authentication unique keys and salts.
 *
 * Change these to different unique phrases! You can generate these using
 * the {Gline https://api.wordpress.org/secret-key/1.1/salt/ WordPress.org secret-key service}.
 *
 * You can change these at any point in time to invalidate all existing cookies.
 * This will force all users to have to log in again.
 *
 * @since 2.6.0
 */
define( 'AUTH_KEY',         'Lg/6x7z-cb_@{_{omc}.8bHbB-k9qF57W1Bu[T]!<r{KC/YCf7!@6<c99$+[`wb`' );
define( 'SECURE_AUTH_KEY',   '|tkgor>^jv{Xo+{s[2MPv>xD_~Y)Fm=<tLTf41Q0zt4xL[93kkhY>5v[x8$+8c`' );
define( 'LOGGED_IN_KEY',    'v_BA3j1w|0S1zxJx_Q(7Rp|+oy1N2d<vxzE9QKw|+@3mGAkhhY7h1k:+e{Mzo-`' );
define( 'NONCE_KEY',        '|})~>z71FCRmxpP00eAun%k|E-(h>Xr|@EKe3glOKK|z7m-j-R1|.cZ2Ou_YQ)=js`' );
define( 'AUTH_SALT',        '015cD@|[#nb980FSF2mt{,.=R|C)QB3xPz0|g6#&cd2JLN7N-in|=sRHP{j=C`' );
define( 'SECURE_AUTH_SALT', '^3_K){Ax0+yee_Fr%6M4.W76+aID{K(h?06Whm@:kx-&#4U1V`|jm=0@#P`' );
define( 'LOGGED_IN_SALT',   'Re=<078LJT_PXr{C;)-347L:[C2Ef+ah+oh1cbHCIE!+8d/1KS~<tu7[n`' );
define( 'NONCE_SALT',       'ybd-[z=HOPKoeCK#/{y2ku_L0/H_O_SwLNo@9n+T]+(+_aq)30FBE,ImSNidAV`' );

Save modified buffer?
  Y Yes
  N No
  C Cancel
```

In the next step, the wordpress files are installed under the Apache Document root:

```
cp -r wordpress/* /var/www/html/
```

To allow the wordpress to use permalinks [2]:

```
sudo nano /etc/httpd/conf/httpd.conf
```

Under a section called <Directory “var/www/html”> the line AllowOverride None is changed to AllowOverride All. This is done so that the wordpress works properly [2].

The file permissions are granted to the apache user since some features in wordpress require the write access. These commands were used to grant file ownership, group ownership and change directory permissions [2].

```
sudo chown -R apache /var/www  
sudo chgrp -R apache /var/www  
sudo chmod 2775 /var/www  
find /var/www -type d -exec sudo chmod 2775 {} \;  
find /var/www -type f -exec sudo chmod 0644 {} \;
```

Wordpress installation

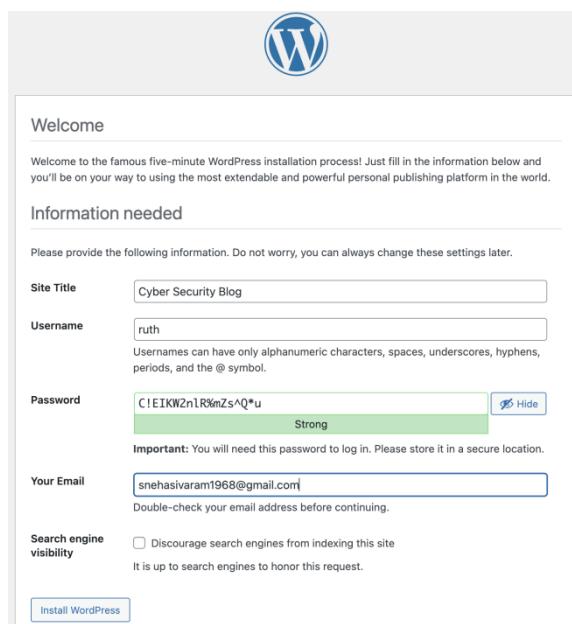
Using chkconfig command to ensure that httpd and database services start at system boot:

```
sudo chkconfig httpd on && sudo chkconfig mariadb on
```

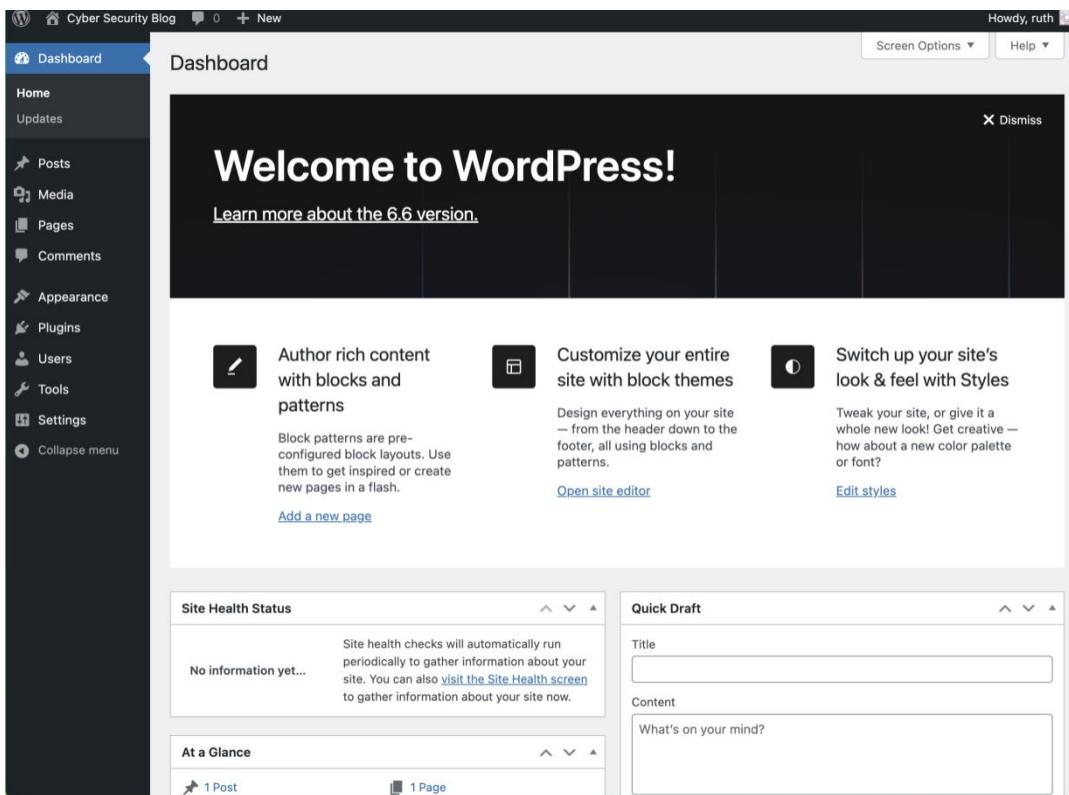
Verifying the database server iand Apache server (httpd) is running:

```
sudo service mariadb status  
sudo service httpd status
```

After these steps, the public DNS address of the instance is accessed through the browser which will take to the wordpress installation page:



After setting up the user and password. Login to the admin page to access the dashboard.



3. Securing and Monitoring the Instance

The most important step in securing the instance always will be regularly updating the operating system and installed packages up to date. Keeping it always updated and patches to fix any possible vulnerabilities is one of the best practices.

And using SSH key pairs is a better and the best option than using password to connect to the instance. This method increases the security by making it difficult for attackers to gain access through brute force attacks.

Firewalld

Firewalld is a dynamic firewall management tool which is can be customized according to the specific needs. It supports varieties of firewall types, rules and settings which makes it useful for protecting an instance [3].

This tool mainly supports linux based distribution and uses zones and services to manage incoming and outgoing traffic and meets various security requirements [3].

Installation of firewalld:

```
sudo yum install firewalld
```

Starting and enabling firewalld:

```
sudo systemctl start firewalld  
sudo systemctl enable firewalld
```

```
[ec2-user@ip-172-31-47-178 ~]$ sudo yum install firewalld  
sudo systemctl start firewalld  
sudo systemctl enable firewalld  
Last metadata expiration check: 6:36:30 ago on Sat Jul 27 10:52:51 2024.  
Dependencies resolved.  
----  
Package           Architecture      Version       Repository     Size  
=====  
Installing:  
firewalld          noarch        1.2.3-1.amzn2023  
Installing dependencies:  
firewalld-filesystem      noarch        1.2.3-1.amzn2023  
gobject-introspection    x86_64       1.73.0-2.amzn2023.0.3  
ipset                x86_64       7.11-1.amzn2023.0.3  
iptables             x86_64       7.1.0-1.amzn2023.0.2  
iptables-libs          x86_64       1.8.8-3.amzn2023.0.2  
iptables-nft            x86_64       1.0.8-2.amzn2023.0.2  
libnetfilter_conntrack  x86_64       1.0.1-19.amzn2023.0.2  
libnftnl              x86_64       1.2.2-2.amzn2023.0.2  
nftables              x86_64       1:1.0.4-3.amzn2023.0.2  
python3-firewall        noarch        1.2.3-1.amzn2023  
python3-gobject-base    x86_64       3.42.2-2.amzn2023.0.3  
python3-gobject-base-noarch x86_64       1:1.0.4-3.amzn2023.0.2  
Installing weak dependencies:  
libcap-ng-python3       x86_64       0.8.2-4.amzn2023.0.2  
----  
Transaction Summary  
=====  
Install 16 Packages  
Total download size: 2.7 M  
Installed size: 11 M  
Is this ok? [y/N] y  
Downloading Packages:  
(1/16): firewalld-filesystem-1.2.3-1.amzn2023.noarch.rpm 137 kB/s | 11 kB 00:00  
(2/16): gobject-introspection-1.73.0-2.amzn2023.0.3.x86_64.rpm 2.5 MB/s | 255 kB 00:00  
(3/16): firewalld-1.2.3-1.amzn2023.noarch.rpm 3.9 MB/s | 452 kB 00:00  
(4/16): ipset-7.11-1.amzn2023.0.3.x86_64.rpm 911 kB/s | 48 kB 00:00  
(5/16): iptables-libs-1.8.8-3.amzn2023.0.2.x86_64.rpm 11 kB/s | 401 kB 00:00  
(6/16): iptables-nft-1.8.8-3.amzn2023.0.2.x86_64.rpm 5.9 MB/s | 183 kB 00:00  
(7/16): libcap-ng-python3-0.8.2-4.amzn2023.0.2.x86_64.rpm 1.3 MB/s | 38 kB 00:00  
(8/16): libnetfilter_conntrack-1.0.8-2.amzn2023.0.2.x86_64.rpm 2.5 MB/s | 58 kB 00:00  
(9/16): libnftnl-1.2.2-1.amzn2023.0.1.x86_64.rpm 7.0 MB/s | 67 kB 00:00  
(10/16): libnftnl-1.0.1-19.amzn2023.0.2.x86_64.rpm 1.3 MB/s | 38 kB 00:00  
(11/16): libnftnl-1.2.2-2.amzn2023.0.2.x86_64.rpm 3.3 MB/s | 84 kB 00:00  
(12/16): nftables-1.0.4-3.amzn2023.0.2.x86_64.rpm 5.9 MB/s | 400 kB 00:00  
(13/16): python3-gobject-base-3.42.2-2.amzn2023.0.3.x86_64.rpm 2.9 MB/s | 178 kB 00:00  
(14/16): python3-firewall-1.2.3-1.amzn2023.noarch.rpm 4.6 MB/s | 357 kB 00:00  
(15/16): python3-nftables-1.0.4-3.amzn2023.0.2.x86_64.rpm 891 kB/s | 18 kB 00:00  
(16/16): python3-gobject-base-noarch-3.42.2-2.amzn2023.0.3.noarch.rpm 2.1 MB/s | 154 kB 00:00  
----  
Total 6.9 MB/s | 2.7 MB 00:00  
Running transaction check  
Transaction check succeeded.  
Running transaction test  
Transaction test succeeded.  
Running transaction  
  Preparing : 1/1  
  Installing : libnftnl-1.2.2-2.amzn2023.0.2.x86_64 1/16  
  Installing : libnftnl-1.0.1-19.amzn2023.0.2.x86_64 2/16  
  Installing : libnetfilter_conntrack-1.0.8-2.amzn2023.0.2.x86_64 3/16
```

Checking the status:

```
sudo firewall-cmd --get-zones
```

Allowing services:

```
sudo firewall-cmd --zone=public --permanent --add-service=http  
sudo firewall-cmd --zone=public --permanent --add-service=https  
sudo firewall-cmd --zone=public --permanent --add-service=ssh
```

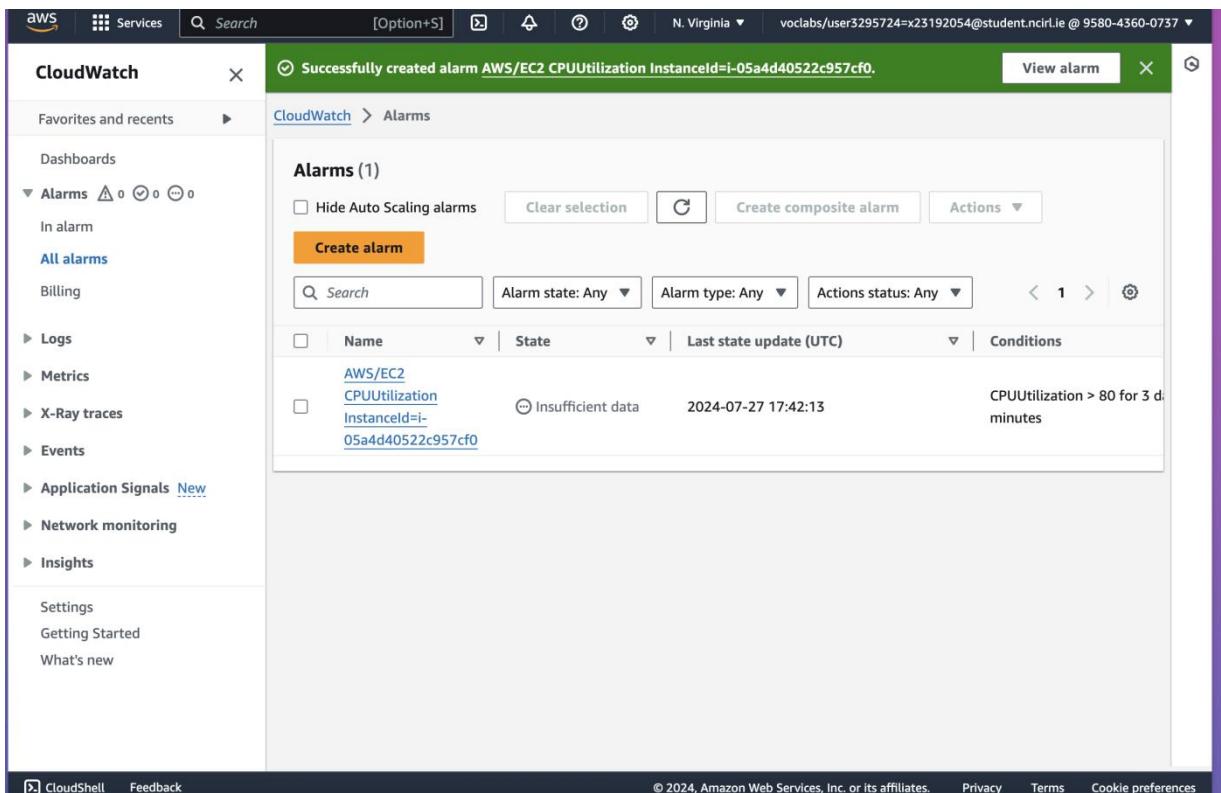
Complete!

```
[ec2-user@ip-172-31-47-178 ~]$ sudo systemctl enable firewalld  
[ec2-user@ip-172-31-47-178 ~]$ sudo systemctl start firewalld  
[ec2-user@ip-172-31-47-178 ~]$ sudo firewall-cmd --zone=public --add-service=http --permanent  
sudo firewall-cmd --zone=public --add-service=https --permanent  
sudo firewall-cmd --zone=public --remove-service=ssh --permanent  
sudo firewall-cmd --reload  
success  
success  
success  
success
```

Cloudwatch

AWS has a tool called cloudwatch which is used to monitor instance's health and can log its behaviour. Cloudwatch can track metrics like CPU utilization, disk I/O, network traffics and is also able to send alerts to the user in case any of the metrics crosses the normal allotted threshold range [4].

Its a great tool for monitoring the health of instance and study its behaviour.



The screenshot shows the AWS CloudWatch Alarms interface. At the top, a green banner indicates "Successfully created alarm AWS/EC2 CPUUtilization InstanceId=i-05a4d40522c957cf0." Below this, the left sidebar lists various monitoring categories: Favorites and recents, Dashboards, Alarms (selected), In alarm, All alarms (highlighted in blue), Billing, Logs, Metrics, X-Ray traces, Events, Application Signals, Network monitoring, Insights, Settings, Getting Started, and What's new. The main content area is titled "Alarms (1)". It includes filters for Hide Auto Scaling alarms, Clear selection, Create composite alarm, and Actions. A search bar and pagination controls are also present. A single alarm entry is listed: "AWS/EC2 CPUUtilization InstanceId=i-05a4d40522c957cf0". The entry shows the alarm state as "Insufficient data", last updated on "2024-07-27 17:42:13", and the condition "CPUUtilization > 80 for 3 minutes".

So this alarm will be monitoring the instance and will report quickly in case anything abnormal occurs.

4. Securing the Wordpress site

Maintaining the security of website requires multiple techniques such as generating a SSL certificate and installing plugins etc. It is important to do this so that the website is secure from any possible attacks.

Generating a SSL/TLS certificate for the website

This SSL/TLS encryption ensures that the data transmitted between the user and the server is completely encrypted without any form of eavesdropping. So here, this tool called certbot was used to generate the certificate.

Certbot is a tool provided by the EEF (Electronic Frontier Foundation) which automates the process

of generating and renewing the ‘Let’s Encrypt’ SSL certificates [5]. Its completely free to use and very useful to have a secure website.

Installation of certbot:

```
sudo dnf install certbot python3-certbot-apache -y
```

Obtaining the SSL certificate:

```
sudo certbot --apache
```

There will series of prompts asking for details like domain name, email address etc. For the public DNS of the instance where the wordpress is installed, a domain name was assigned to it using the online platform GoDaddy which was used for generating the SSL certificate.

Plugins for Wordpress:

1. Limit Login Attempts Reloaded

This plugin limits the number of login attempts from one single IP address thereby preventing brute force attacks [6]. Its basically easy to install and setup in the wordpress plugin section. It can be configured to set the number of allowed attempts and other metrics [6].

2. Login Lockdown

This plugin records IP addresses and timestamps of failed login attempts and locks down login if there are too many bad login attempts [7]. This can be easily set up as plugin and configured to set the number of login attempts and lockdown duration as well [7].

3. Akismet

This plugin was preinstalled in the wordpress, when the site was first hosted. It is known to be a powerful anti-spam plugin that checks comments and contact form submissions against a global database of spam [8]. This plugin is activated by obtaining the API key from the official website.

4. Solid Security Basic

Its a great plugin which has a wide range of features such as two factor authentication, file change detection and security logs [9]. It can be customized to specific needs and the 2FA authentication can be assigned to specific roles which are required as well [9]. It is easy to install and activate.

5. Updraft plus

Its a backup plugin that allows to schedule backups at specific times. It is very useful and greatly helps in case anything happens to website [10]. It also has the features of storing backups in remote and safer locations [10].

6. Wordfence Security

Its a very useful plugin which provides complete security solutions such as a firewall, malware scanner, traffic and login attempts monitoring etc. It has a separate dashboard and has a wide variety

of features that can be used. It is easy to install and set up the security features [11].

5. Testing of wordpress

After implementing all these plugins, it is important to always monitor the website. Since the plugins may not always guarantee a hundred percent safety. So, some of the important tools were used as part of testing the website to check for any possible issues that could be noted down.

Here, two tools were mainly chosen: Nikto and Nmap. These two tools were used in a virtual machine Kali Linux, and was able to yield some important results regarding the website.

Nikto:

It is a powerful web server scanning tool that tests for various security issues, outdated software, insecure files and other problems [12].

The command to use nikto in kali linux:

```
nikto -h http://54.144.123.185
```

The following results were found:

```
(ruth㉿ruth):[~]$ nikto -h http://54.144.123.185
- Nikto v2.5.0
+ Target IP:      54.144.123.185
+ Target Hostname: 54.144.123.185
+ Target Port:     80
+ Start Time:    2024-07-28 03:23:45 (GMT-7)

+ Server: Apache/2.4.61 (Amazon Linux) OpenSSL/3.0.8
+ /: The anti-clickjacking X-Frame-Options header is not present. See: https://developer.mozilla.org/en-US/docs/Web/HTTP/Headers/X-Frame-Options
+ /: Drupal Link header found with value: <http://54.144.123.185/wp-json/>; rel="https://api.w.org/". See: https://www.drupal.org/
+ /: The X-Content-Type-Options header is not set. This could allow the user agent to render the content of the site in a different fashion to the MIME type . See: https://www.netsparker.com/web-vulnerability-scanner/vulnerabilities/missing-content-type-header/
+ /XL3Eu809.: Uncommon header 'x-redirect-by' found, with contents: WordPress.
+ /robots.txt: contains 2 entries which should be manually viewed. See: https://developer.mozilla.org/en-US/docs/Glossary/Robots.txt
+ /: Web Server returns a valid response with junk HTTP methods which may cause false positives.
+ /: HTTP TRACE method is active which suggests the host is vulnerable to XST. See: https://owasp.org/www-community/attacks/Cross\_Site\_Tracing
+ /icons/: Directory indexing found.
+ /icons/README: Apache default file found. See: https://www.vntweb.co.uk/apache-restricting-access-to-iconsreadme/
+ /wp-links-opml.php: This WordPress script reveals the installed version.
+ /license.txt: License file found may identify site software.
+ /wp-app.log: Wordpress' wp-app.log may leak application/system details.
+ /wordpress/wp-app.log: Wordpress' wp-app.log may leak application/system details.
+ /: A Wordpress installation was found.
+ /wordpress/: A Wordpress installation was found.
+ /wp-login.php?action=register: Cookie wordpress_test_cookie created without the httponly flag. See: https://developer.mozilla.org/en-US/docs/Web/HTTP/Cookies
+ /wp-login.php: Cookie PHPSESSID created without the httponly flag. See: https://developer.mozilla.org/en-US/docs/Web/HTTP/Cookies
+ /wp-login.php: Wordpress login found.
+ 8913 requests: 0 error(s) and 18 item(s) reported on remote host
+ End Time:        2024-07-28 03:53:57 (GMT-7) (1812 seconds)

+ 1 host(s) tested
```

It was able to detect HTTP Trace method enabled which must be disabled to prevent any Cross-site Scripting attack. And files containing sensitive data related to the website were detected which must be restricted of any access.

The website can be protected by regularly monitoring and testing with tools to find any possible issues.

Nmap:

Nmap is a powerful network scanning tool which is used to discover open hosts, ports and services to detect any possible vulnerabilities [13]. It can also detect the software versions and is widely used by security professionals for network scanning.

The command to use Nmap in Kali Linux is:

```
nmap -sV -Pn 54.144.123.185
```

For scanning vulnerabilities:

```
nmap --script vuln 54.144.123.185
```

The following results were generated:

```
(ruth㉿ruth) ~ $ nmap -sV -Pn 54.144.123.185
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-07-27 17:13 PDT
Nmap scan report for ec2-54-144-123-185.compute-1.amazonaws.com (54.144.123.185)
Host is up (0.089s latency).
Not shown: 996 filtered tcp ports (no-response)
PORT      STATE SERVICE VERSION
22/tcp    open  ssh    OpenSSH 8.7 (protocol 2.0)
53/tcp    open  domain dnsmasq 2.84rc2
80/tcp    open  http   Apache httpd 2.4.61 ((Amazon Linux) OpenSSL/3.0.8)
443/tcp   open  ssl/http Apache httpd 2.4.61 ((Amazon Linux) OpenSSL/3.0.8)

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 23.18 seconds

(ruth㉿ruth) ~ $
```

```
(ruth㉿ruth) ~ $ nmap --script vuln 54.144.123.185
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-07-27 17:19 PDT
Pre-scan script results:
| broadcast-avahi-dos:
|_ Discovered hosts:
|   224.0.0.251
| After NULL UDP avahi packet DoS (CVE-2011-1002).
|_ Hosts are all up (not vulnerable).
Nmap scan report for ec2-54-144-123-185.compute-1.amazonaws.com (54.144.123.185)
Host is up (0.088s latency).
Not shown: 996 filtered tcp ports (no-response)
PORT      STATE SERVICE
22/tcp    open  ssh
53/tcp    open  domain
80/tcp    open  http
| http-trace: TRACE is enabled
| http-stored-xss: Couldn't find any stored XSS vulnerabilities.
| http-dombased-xss: Couldn't find any DOM based XSS.
| http-csrf: Couldn't find any CSRF vulnerabilities.
| http-enum:
|   /wp-login.php: Possible admin folder
|   /wp-json: Possible admin folder
|   /robots.txt: Robots file
|   /: Wordpress version: 6.6.1
|   /feed/: Wordpress version: 6.6.1
|   /wp-includes/images/rss.png: Wordpress version 2.2 found.  []
|   /wp-includes/js/jquery/suggest.js: Wordpress version 2.5 found.
|   /wp-includes/images/blank.gif: Wordpress version 2.6 found.
|   /wp-includes/js/comment-reply.js: Wordpress version 2.7 found.
|   /wp-login.php: Wordpress login page.
|   /wp-admin/upgrade.php: Wordpress login page.
|   /0/: Potentially interesting folder
|   /icons/: Potentially interesting folder w/ directory listing
443/tcp   open  https
| http-trace: TRACE is enabled
| http-enum:
|   /wp-login.php: Possible admin folder
|   /wp-json: Possible admin folder
|   /robots.txt: Robots file
```

```
|_http-dombased-xss: Couldn't find any DOM based XSS.
|_http-csrf: Couldn't find any CSRF vulnerabilities.
| http-enum:
|   /wp-login.php: Possible admin folder
|   /wp-json: Possible admin folder
|   /robots.txt: Robots file
|   /: WordPress version: 6.6.1
|   /feed/: Wordpress version: 6.6.1
|   /wp-includes/images/rss.png: Wordpress version 2.2 found.
|   /wp-includes/js/jquery/suggest.js: Wordpress version 2.5 found.
|   /wp-includes/images/blank.gif: Wordpress version 2.6 found.
|   /wp-includes/js/comment-reply.js: Wordpress version 2.7 found.
|   /wp-login.php: Wordpress login page.
|   /wp-admin/upgrade.php: Wordpress login page.
|   /0/: Potentially interesting folder
|_ /icons/: Potentially interesting folder w/ directory listing
443/tcp open https
|_http-trace: TRACE is enabled
| http-enum:
|   /wp-login.php: Possible admin folder
|   /wp-json: Possible admin folder
|   /robots.txt: Robots file
|   /: WordPress version: 6.6.1
|   /feed/: Wordpress version: 6.6.1
|   /wp-includes/images/rss.png: Wordpress version 2.2 found.
|   /wp-includes/js/jquery/suggest.js: Wordpress version 2.5 found.
|   /wp-includes/images/blank.gif: Wordpress version 2.6 found.
|   /wp-includes/js/comment-reply.js: Wordpress version 2.7 found.
|   /wp-login.php: Wordpress login page.
|   /wp-admin/upgrade.php: Wordpress login page.
|   /0/: Potentially interesting folder
|_ /icons/: Potentially interesting folder w/ directory listing
|_http-stored-xss: Couldn't find any stored XSS vulnerabilities.
|_http-dombased-xss: Couldn't find any DOM based XSS.
|_http-csrf: Couldn't find any CSRF vulnerabilities.
```

Nmap done: 1 IP address (1 host up) scanned in 295.01 seconds

```
└─$
```

It was able to discover the open ports: 22, 53, 80 and 443. And it also detected that TRACE was enabled on both port 80 and port 443. Some of the admin folders were also detected.

These issues can be resolved by restricting access to certain ports, and deleting unwanted files. And installing more secure plugins as well.

It is important to always monitor the wordpress and instance and check for any unusual behaviour and the software must be always be kept up to date. These are the best practices that could be followed to maintain the security of Instance and the web application. To conclude, more advanced tools can be used to maintain the security of both very well.

REFERENCES:

- [1] AWS, “Tutorial: Install a LAMP server on AL2023,” 2023. [Online]. Available: <https://docs.aws.amazon.com/linux/al2023/ug/ec2-lamp-amazon-linux-2023.html#test-lamp-server-2023> [Accessed on: July 10, 2024].
- [2] AWS, “ Tutorial: Host a WordPress blog on AL2023,” 2023. [Online]. Available: <https://docs.aws.amazon.com/linux/al2023/ug/hosting-wordpress-aml-2023.html> [Accessed on: July 10, 2024]
- [3] Firewalld.org, “Firewalld,” 2023. [Online]. Available: <https://firewalld.org/> [Accessed on: July 10, 2024]
- [4] AWS, “What is Amazon CloudWatch?,” 2021. [Online]. Available: <https://docs.aws.amazon.com/AmazonCloudWatch/latest/monitoring/WhatIsCloudWatch.html> [Accessed on: July 10, 2024]
- [5] EFF, “Certbot.”. [Online]. Available: <https://certbot.eff.org/> [Accessed on: July 10, 2024]
- [6] Wordpress, “Limit Login Attempts Reloaded,” 2024. [Online]. Available: <https://wordpress.org/plugins/limit-login-attempts-reloaded/> [Accessed on: July 10, 2024]
- [7] Wordpress, “Login Lockdown and Protection,” 2024. [Online]. Available: <https://wordpress.org/plugins/login-lockdown/> [Accessed on: July 10, 2024]
- [8] Wordpress, “Akismet Anti-spam: Spam Protection,” 2024. [Online]. Available: <https://wordpress.org/plugins/akismet/> [Accessed on: July 11, 2024]
- [9] Wordpress, “Solid Security – Password, Two Factor Authentication, and Brute Force Protection,” 2024. [Online]. Available: <https://wordpress.org/plugins/better-wp-security/> [Accessed on: July 11, 2024]
- [10] Wordpress, “UpdraftPlus: WP Backup & Migration Plugin,” 2024. [Online]. Available: <https://wordpress.org/plugins/updraftplus/> [Accessed on: July 11, 2024]
- [11] Wordpress, “Wordfence Security – Firewall, Malware Scan, and Login Security,” 2024. [Online]. Available: <https://wordpress.org/plugins/wordfence/> [Accessed on: July 11, 2024]
- [12] CIRT.net, “Nikto 2.5,” 2024. [Online]. Available: <https://www.cirt.net/Nikto2> [Accessed on: July 11, 2024]
- [13] Nmap.org, “Nmap 7.95,” 2024. [Online]. Available: <https://nmap.org/> [Accessed on: July 11, 2024]

APPENDIX

Setting up MariaDB

```
You already have your root account protected, so you can safely answer 'n'.
```

```
[Change the root password? [Y/n] y  
[New password:  
[Re-enter new password:  
Password updated successfully!  
Reloading privilege tables..  
... Success!
```

By default, a MariaDB installation has an anonymous user, allowing anyone to log into MariaDB without having to have a user account created for them. This is intended only for testing, and to make the installation go a bit smoother. You should remove them before moving into a production environment.

```
[Remove anonymous users? [Y/n] y  
... Success!
```

Normally, root should only be allowed to connect from 'localhost'. This ensures that someone cannot guess at the root password from the network.

```
[Disallow root login remotely? [Y/n] y  
... Success!
```

By default, MariaDB comes with a database named 'test' that anyone can access. This is also intended only for testing, and should be removed before moving into a production environment.

```
[Remove test database and access to it? [Y/n] y  
- Dropping test database...  
... Success!  
- Removing privileges on test database...  
... Success!
```

Reloading the privilege tables will ensure that all changes made so far will take effect immediately.

```
[Reload privilege tables now? [Y/n] y  
... Success!
```

Cleaning up...

All done! If you've completed all of the above steps, your MariaDB installation should now be secure.

Thanks for using MariaDB!

```
[ec2-user@ip-172-31-92-50 ~]$ █
```

```
[ec2-user@ip-172-31-92-50 ~]$ dnf install wget php-mysqld httpd php-fpm php-mysqli mariadb105-server php-json php php-devel -y  
Error: This command has to be run with superuser privileges (under the root user on most systems).  
[ec2-user@ip-172-31-92-50 ~]$ tar -xzf latest.tar.gz  
[ec2-user@ip-172-31-92-50 ~]$ sudo systemctl start mariadb httpd  
[ec2-user@ip-172-31-92-50 ~]$ mysql -u root -p  
Enter password:  
Welcome to the MariaDB monitor. Commands end with ; or \g.  
Your MariaDB connection id is 13  
Server version: 10.5.23-MariaDB MariaDB Server  
  
Copyright (c) 2000, 2018, Oracle, MariaDB Corporation Ab and others.  
Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.  
MariaDB [(none)]> CREATE USER 'clouduser'@'localhost' IDENTIFIED BY 'cloud123';  
Query OK, 0 rows affected (0.004 sec)  
MariaDB [(none)]> CREATE DATABASE `wordpress-db`;  
Query OK, 1 row affected (0.000 sec)  
MariaDB [(none)]> CREATE USER 'clouduser'@'localhost' IDENTIFIED BY cloud123;  
ERROR 1064 (42000): You have an error in your SQL syntax; check the manual that corresponds to your MariaDB server version for the right syntax to use near 'cloud123' at line 1  
MariaDB [(none)]> CREATE USER 'clouduser'@'localhost' IDENTIFIED BY 'cloud123';  
ERROR 1396 (HY000): Operation CREATE USER failed for 'clouduser'@'localhost'  
MariaDB [(none)]> GRANT ALL PRIVILEGES ON `wordpress-db`.* TO "clouduser"@"localhost";  
Query OK, 0 rows affected (0.001 sec)  
MariaDB [(none)]> FLUSH PRIVILEGES;  
Query OK, 0 rows affected (0.001 sec)  
MariaDB [(none)]> exit  
Bye
```

Setting up Cloudwatch

CloudWatch > Alarms > Create alarm

Step 1 Specify metric and conditions

Step 2 Configure actions

Step 3 Add name and description

Step 4 Preview and create

Specify metric and conditions

[Create alarm using Infrastructure as Code - new](#) [View code](#)

Metric

Graph
This alarm will trigger when the blue line goes above the red line for 3 datapoints within 15 minutes.

Percent

Namespace AWS/EC2

Metric name CPUUtilization

InstanceId i-05a4d40522c957cf0

Instance name cloudstack

Statistic [Average](#)

Period 5 minutes

Conditions

Threshold type

Static [?](#)
Use a value as a threshold

Anomaly detection [?](#)
Use a band as a threshold

Whenever CPUUtilization is...
Define the alarm condition.

Greater [?](#)
> threshold

Greater/Equal [?](#)
>= threshold

Lower/Equal [?](#)
<= threshold

Lower [?](#)
< threshold

than... [?](#)
Define the threshold value.

80

Must be a number

► Additional configuration

[CloudWatch](#) > [Alarms](#) > Create alarm

Step 1
[Specify metric and conditions](#)

Step 2
[Configure actions](#)

Step 3
Add name and description

Step 4
Preview and create

Configure actions

Notification

Alarm state trigger
Define the alarm state that will trigger this action.

In alarm
The metric or expression is outside of the defined threshold.

OK
The metric or expression is within the defined threshold.

Insufficient data
The alarm has just started or not enough data is available.

Send a notification to the following SNS topic
Define the SNS (Simple Notification Service) topic that will receive the notification.

Select an existing SNS topic

Create new topic

Use topic ARN to notify other accounts

Send a notification to...

Only topics belonging to this account are listed here. All persons and applications subscribed to the selected topic will receive notifications.

Add notification

SSL/TLS certificate generation

```
lines 1-21/21 [END]
[ec2-user@ip-172-31-92-50 ~]$ sudo dnf install certbot python3-certbot-apache -y
Last metadata expiration check: 1:16:30 ago on Mon Jul 22 09:30:59 2024.
Dependencies resolved.
=====
Package          Arch    Version           Repository      Size
=====
Installing:
certbot           noarch  2.6.0-4.amzn2023.0.1   amazonlinux   49 k
python3-certbot-apache  noarch  2.6.0-4.amzn2023.0.1   amazonlinux   287 k
Installing dependencies:
augeas-libs        x86_64  1.13.0-1.amzn2023.0.2   amazonlinux  408 k
fontawesome-fonts  noarch  1:4.7.0-11.amzn2023.0.2   amazonlinux  285 k
mod_ssl           x86_64  1:2.4.59-2.amzn2023   amazonlinux  112 k
python3-acme        noarch  2.6.0-4.amzn2023.0.1   amazonlinux  161 k
python3-augeas       noarch  1.1.0-10.amzn2023   amazonlinux  34 k
python3-certbot     noarch  2.6.0-4.amzn2023.0.1   amazonlinux  677 k
python3-configargparse noarch  1.7-1.amzn2023   amazonlinux  45 k
python3-josepy       noarch  1.13.0-6.amzn2023   amazonlinux  61 k
python3-parsedatetime noarch  2.6-18.amzn2023   amazonlinux  88 k
python3-pyOpenSSL    noarch  21.0.0-1.amzn2023.0.2   amazonlinux  97 k
python3-pyrfc339     noarch  1.1-16.amzn2023   amazonlinux  19 k
sscg              x86_64  3.0.3-76.amzn2023   amazonlinux  45 k
Installing weak dependencies:
python-josepy-doc   noarch  1.13.0-6.amzn2023   amazonlinux  20 k
=====
Transaction Summary
=====
Install 15 Packages

Total download size: 2.2 M
Installed size: 9.5 M
Downloading Packages:
(1/15): certbot-2.6.0-4.amzn2023.0.1.noarch.rpm 420 kB/s | 49 kB  00:00
(2/15): fontawesome-fonts-4.7.0-11.amzn2023.0.2.1.4 MB/s | 285 kB  00:00
(3/15): augeas-libs-1.13.0-1.amzn2023.0.2.x86_64 2.6 MB/s | 408 kB  00:00
(4/15): python-josepy-doc-1.13.0-6.amzn2023.noa 725 kB/s | 20 kB  00:00
(5/15): mod_ssl-2.4.59-2.amzn2023.x86_64.rpm 1.4 MB/s | 112 kB  00:00
(6/15): python3-augeas-1.1.0-10.amzn2023.noarch 410 kB/s | 34 kB  00:00
(7/15): python3-acme-2.6.0-4.amzn2023.0.2.noarch 1.2 MB/s | 146 kB  00:00
(8/15): python3-configargparse-1.7-1.amzn2023   1.2 MB/s | 45 kB  00:00
(9/15): python3-certbot-apache-2.6.0-4.amzn2023 4.0 MB/s | 287 kB  00:00
(10/15): python3-certbot-2.6.0-4.amzn2023.0.1.n.4.6 MB/s | 677 kB  00:00
(11/15): python3-josepy-1.13.0-6.amzn2023.noarch 1.8 MB/s | 61 kB  00:00
(12/15): python3-pyrfc339-1.1-16.amzn2023.noar 775 kB/s | 19 kB  00:00
(13/15): python3-parsedatetime-2.6-10.amzn2023. 1.4 MB/s | 88 kB  00:00
(14/15): sschg-3.0.3-76.amzn2023.x86_64.rpm 1.9 MB/s | 45 kB  00:00
(15/15): python3-pyOpenSSL-21.0.0-1.amzn2023.0. 1.3 MB/s | 92 kB  00:00
=====
Total download size: 2.2 M
Running transaction check
Transaction check succeeded.
Running transaction test
Transaction test succeeded.
Running transaction
Preparing : python3-pyrfc339-1.1-16.amzn2023.noarch 1/15
Installing : python3-pyOpenSSL-21.0.0-1.amzn2023.0.2.noarch 2/15
Installing : sschg-3.0.3-76.amzn2023.x86_64 3/15
Installing : mod_ssl-2.4.59-2.amzn2023.x86_64 4/15
Installing : python3-parsedatetime-2.6-10.amzn2023.noarch 5/15
Installing : python3-configargparse-1.7-1.amzn2023.noarch 6/15
Installing : python-josepy-doc-1.13.0-6.amzn2023.noarch 7/15
```

```
[ec2-user@ip-172-31-92-50 ~]$ sudo certbot --apache -v -d ruthisnotadev.com
Saving debug log to /var/log/letsencrypt/letsencrypt.log
Plugins selected: Authenticator apache, Installer apache
Requesting a certificate for ruthisnotadev.com
Performing the following challenges:
http-01 challenge for ruthisnotadev.com
Waiting for verification...
Cleaning up challenges

Successfully received certificate.
Certificate is saved at: /etc/letsencrypt/live/ruthisnotadev.com/fullchain.pem
Key is saved at: /etc/letsencrypt/live/ruthisnotadev.com/privkey.pem
This certificate expires on 2024-10-20.
These files will be updated when the certificate renews.
Certbot has set up a scheduled task to automatically renew this certificate in the background.
```

```
Deploying certificate
Created an SSL vhost at /etc/httpd/conf/httpd-le-ssl.conf
Deploying Certificate to VirtualHost /etc/httpd/conf/httpd-le-ssl.conf
Enabling site /etc/httpd/conf/httpd-le-ssl.conf by adding Include to root configuration
Successfully deployed certificate for ruthisnotadev.com to /etc/httpd/conf/httpd-le-ssl.conf
Redirecting vhost in /etc/httpd/conf/httpd.conf to ssl vhost in /etc/httpd/conf/httpd-le-ssl.conf
Congratulations! You have successfully enabled HTTPS on https://ruthisnotadev.com
Subscribe to the EFF mailing list (email: snehasivaram1968@gmail.com).

-----
If you like Certbot, please consider supporting our work by:
* Donating to ISRG / Let's Encrypt: https://letsencrypt.org/donate
* Donating to EFF: https://eff.org/donate-le
-----
```

```
[ec2-user@ip-172-31-92-50 ~]$ █
```

The certificate:

```
</Directory>
ErrorLog /var/log/httpd/your-domain-error.log
CustomLog /var/log/httpd/your-domain-access.log combined

Include /etc/letsencrypt/options-ssl-apache.conf
SSLCertificateFile /etc/letsencrypt/live/ruthisnotadev.com/fullchain.pem
SSLCertificateKeyFile /etc/letsencrypt/live/ruthisnotadev.com/privkey.pem
</VirtualHost>
</IfModule>
[ec2-user@ip-172-31-66-245 ~]$ cd /etc/letsencrypt/live/ruthisnotadev.com/fullchain.pem
[ec2-user@ip-172-31-66-245 ~]$ cd /etc/letsencrypt/live/ruthisnotadev.com/privkey.pem
[bash: cd: /etc/letsencrypt/live/ruthisnotadev.com/fullchain.pem: Permission denied]
[ec2-user@ip-172-31-66-245 ~]$ sudo cat /etc/letsencrypt/live/ruthisnotadev.com/fullchain.pem
-----BEGIN CERTIFICATE-----
MIIDnDCCAYKgAwIBAgISaYRvBN7WEUE10hAVXG4+HQWnAaGCCgSM49BAMMDIx
CzABgNVBAYTATVTRMyrFAYDVQKEW1ZxQncyBfMjnaBwgXJAYBgNVBAMTEZxI
NTAEwFwYwNDAMjIaFwYwNDMjIaFwYwNDMjIaFwYwNDMjIaFwYwNDMjIaFwYwNDMjIaFw
dGhpC25vdFkXYtZyU2tHwleWtHkZ1zJ0AQYIKoZ1zJ0DAQcQgAEznKZTAY
qN4Qx9sIOTQyEVSp5Vg/3M0uLxxYYMdavz85CEu0Ps4crsRlDEB15z2xH
IY2zIwDfEzJ0PwRmYjQ0QAvRmYjQ0QAvRmYjQ0QAvRmYjQ0QAvRmYjQ0QAvRmYjQ0
BgfB0oDAOY1KwVBQ0JhnaLwADYvR87AGh/BAwADadBqVHh4FgQuo8s9s7Yq5
1+s7VxK84q1.7zr6s1vEwlewDVr0/BBwpFaAUuytfzwh750t+E+9+rLMTGc1s1w8w
VQY1KwVBQ0JhNAQEESTBHMCEGCCsGAOUFBzAHbhVdHhw0i8zTUUhbs5Zwsjci5v
cmcwigY1KwVBQ0JhMAKGfmhd8dHA61y91NS5plmx1bmhyLy9mZy8wMyDVR8RCaw
KoiRcnV9e01z9w9YwRl1s5jb22CFxdy5vxRoqXhul3RhzGV2leNbvtATBq9w
HSAEEDAKMAG0mBeDAECATCAQYOClsAQGBIn1KCBA1EgfcEgFQA8gB3AD8X89/X
1kdY1B1LHS=DRLtK0d/H4vq68G/KIx+gRuAAABPaQSMwAAAQDAAegRghA05
rw18fTlOfv6AxmzV1IXtDfEl1fgYKkFunDg0ogOh1CA1eA9JwH9SUdkyFmVT91pkvg
MFy97XFDfNgQAn3r3CsyeAdwAzmBbxCTDwz4wNgNeK257g24ozPkuUo7385
PAw+XfL0uEYy0oWwM0Zt34tCQCD/C/uwQXSp0g0xrxz2/EOOBdzkzH+nHw
Unhmf9hsPH21AMC94+rFlx1M2Hm0mlnfEd41Nz/FHN2Hmhkk1deJ071MA0gCcG
SM49BAMDA2gPmuQH0dfLHz0+5nItfYEdF40gea3eQ748b0ywnD03NzAEHrPew
HtsC3a+k6GySreI918CMAR4IBUt+v6lppcIIln+1lq1tODVGOM/4mUnv1
yxuCNICCInhdKLCxhNgn=-----END CERTIFICATE-----
```

```
-----BEGIN CERTIFICATE-----
```

```
-----BEGIN PRIVATE KEY-----
```

```
MIGHAgEAMBMBGByqGSM49AgEGCCqGSM49AwEHBG0wawIBAQqaSMSX7tMKR70AzbJ
83J1cdaXc5T4CYCHDj2e0ZumLShRANCAAT0eR1Mctio3hDh2whBoJi68s9I9JWD
/cMfxNTGRhx1rlPTkIs7RVU+zhyuxEt0QG2Xm3bEgchjMYdPFFOHBIW
```

```
-----END PRIVATE KEY-----
```

```
[ec2-user@ip-172-31-66-245 ~]$ █
```

```
[ec2-user@ip-172-31-66-245 ~]$ sudo cat /etc/letsencrypt/live/ruthisnotadev.com/privkey.pem
```

The Certificate Details:

Certificate Viewer: ruthisnotadev.com X

General Details

Issued To

Common Name (CN)	ruthisnotadev.com
Organisation (O)	<Not part of certificate>
Organisational Unit (OU)	<Not part of certificate>

Issued By

Common Name (CN)	E5
Organisation (O)	Let's Encrypt
Organisational Unit (OU)	<Not part of certificate>

Validity Period

Issued On	Saturday 27 July 2024 at 23:39:29
Expires On	Friday 25 October 2024 at 23:39:28

SHA-256 Fingerprints

Certificate	73e5697b793fe7077b8d1ade2c85b79f06dadf844fa87d72813c1 91316393c31
Public key	7ea58cc7c0a7873082cb2bbb55040aa6cf8fba2a751a6adf62a40 edd89ee993b

Wordpress plugins

Wordfence

The screenshot shows the Wordfence dashboard with the following key elements:

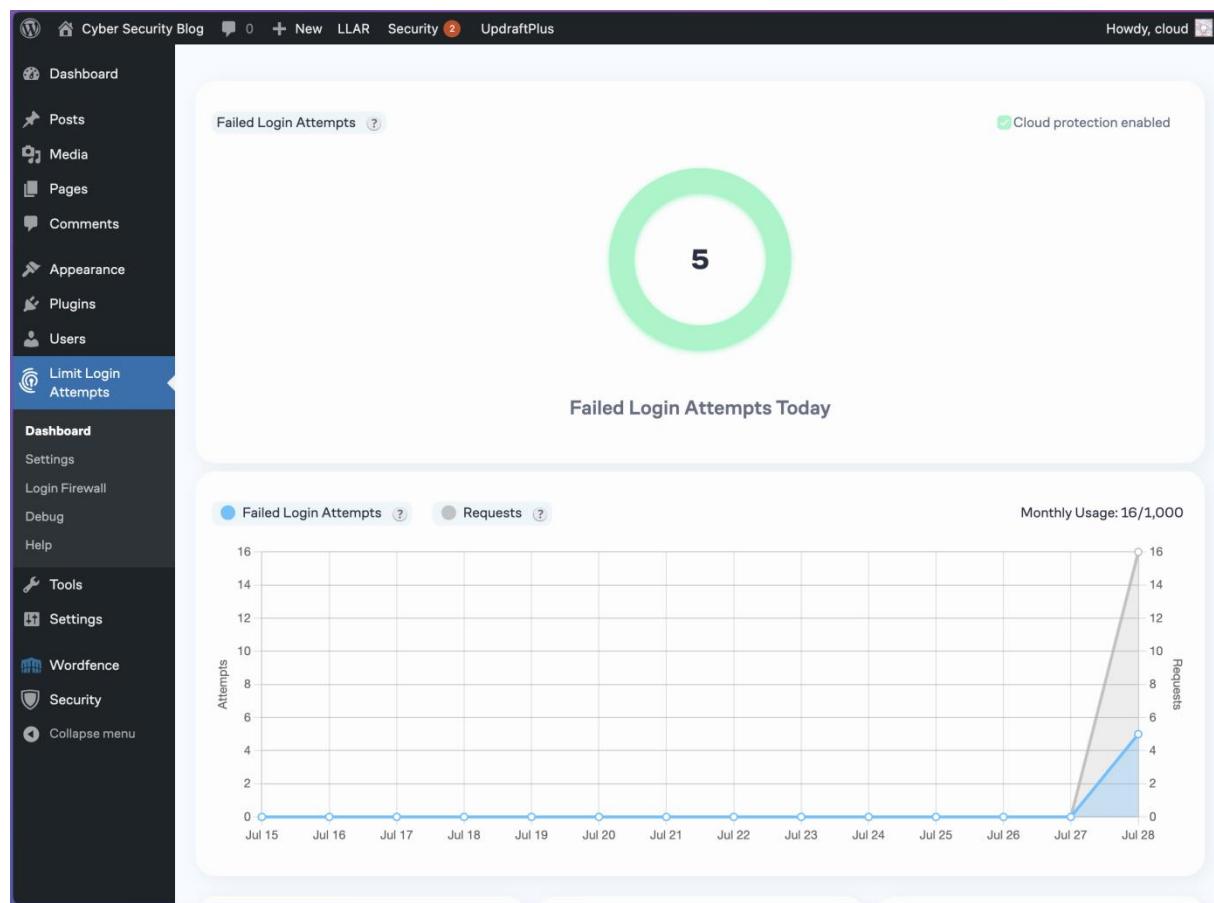
- Header:** Shows "Cyber Security Blog" and "Howdy, cloud". A message "You're still goin' strong" is displayed.
- Left Sidebar:** Includes links for Dashboard, Posts, Media, Pages, Comments, Appearance, Plugins, Users, Tools, Settings, and Wordfence (which is selected).
- Top Bar:** A yellow banner with the text "To make your site as secure as possible, take a moment to optimize the Wordfence Web Application Firewall:" and a "CLICK HERE TO CONFIGURE" button. Below it is a "DISMISS" button and a note about setup help.
- Middle Section:** A large blue header box says "Wordfence Protection Activated". It contains two circular progress indicators: one for "Firewall" at 48% and another for "Scan" at 60%. Below each indicator are "Manage Firewall" and "Manage Scan" buttons respectively.
- Right Column:** A section titled "Premium Protection Disabled" encourages upgrading to Premium. It includes a "UPGRADE TO PREMIUM" button and a "LEARN MORE" button.
- Bottom Section:** A "Notifications" box shows "No notifications received". To its right is a "Wordfence Central Status" box with a gear icon and the text: "Wordfence Central allows you to manage Wordfence on multiple sites from one location. It makes security monitoring and configuring Wordfence easier."

Firewall enabled

The screenshot shows the Firewall Options page with the following key elements:

- Header:** A yellow banner with the text "To make your site as secure as possible, take a moment to optimize the Wordfence Web Application Firewall:" and a "CLICK HERE TO CONFIGURE" button. Below it is a "DISMISS" button and a note about setup help.
- Section Headers:** "Firewall Options" and "Learn more about the Firewall".
- Four Protection Levels:** Each represented by a circle with a percentage:
 - Web Application Firewall:** 35% (red outline)
 - Firewall Rules: Community:** 70% (orange outline)
 - Real-Time IP Blocklist: Disabled:** 0% (grey outline)
 - Brute Force Protection:** 100% (green outline)
- Basic Firewall Options:** A table with three columns:
 - Web Application Firewall Status:** Enabled and Protecting
 - Protection Level:** Basic WordPress Protection: The plugin will load as a regular plugin after WordPress has been loaded, and while it can block many malicious requests, some vulnerable plugins or WordPress itself may run vulnerable code before all plugins are loaded.
 - Real-Time IP Blocklist:** Premium Feature: This feature blocks all traffic from IPs with a high volume of recent malicious activity using Wordfence's real-time blocklist.

Limit Login Attempts



The screenshot shows the 'Limit Login Attempts Reloaded Cloud App' settings page. The left sidebar has a dark theme with various menu items. The main area contains configuration options and promotional content.

Setup Code: Edit (4)

A number of attempts the user has to enter their credentials correctly.

Allowed Retries: 4

Lockout Interval: 20

Initial lockout interval in minutes. It will temporarily increase automatically after each consecutive lockout.

Block XML-RPC: on

Block all login attempts made against XML-RPC from public IPs.

Auto IP Blocklist: on

Automatically add IPs to block-list after 3 lockouts.

Why Use Our Premium Cloud App?

- Absorb site load caused by attacks**
- Use intelligent IP denial/unblocking technology**
- Sync the allow/deny/pass lists between multiple domains**

Upgrade

Solid Security

The screenshot shows the 'Global Settings' step of the Solid Security setup process. On the left, a dark sidebar lists various WordPress and Solid Security settings. The 'Security' section is currently selected. The main area displays 'Global Settings' with a sub-section titled 'Authorized IPs'. It shows a list of authorized IP addresses: '51.37.236.204'. A text input field below allows for entering more IP addresses, with a placeholder 'Enter a list of IP addresses that should not be locked out by Solid Security.' A button labeled 'Authorize my IP address' is present. Another section, 'IP Detection', includes 'PROXY DETECTION' with a dropdown menu set to 'Security Check Scan (Recommended)'. A note explains that this setting configures how Solid Security identifies visitor IP addresses. A 'Check IP' button and a 'Detected IP: 51.37.236.204' indicator are also shown. A 'Next' button is located in the top right corner.

Enabling two factor authentication

The screenshot shows the 'Features' step of the Solid Security setup process. The left sidebar remains the same. The main area displays 'Features' with a sub-section titled 'Login Security'. It features a toggle switch for 'Two-Factor' authentication, which is turned on. A descriptive text box explains that Two-Factor Authentication increases security by requiring an additional code along with the username and password. Buttons for 'Undo Changes' and 'Next' are at the bottom right. A navigation bar at the bottom indicates the current step: '3 Features'.

The user groups

This screenshot shows the 'User Groups' section of the WordPress Security plugin. The left sidebar is dark-themed and includes links for Dashboard, Posts, Media, Pages, Comments, Appearance, Plugins, Users, Limit Login Attempts, Tools, Settings, Wordfence, and Security (which is highlighted). The main content area has a purple header with a back button and a 'User Groups 4' title. A sub-header 'Default User Groups' with a note about managing security settings per group is followed by tabs for Administrators, Editors, Authors, Contributors, Subscribers, Everybody Else, and a plus sign. Below this is a 'Features' tab, which is selected, and an 'Edit Group' tab. The 'Global Settings' section contains a toggle for 'Manage Solid Security' (on) and a note about allowing users to manage solid security. The 'Security Dashboard' section contains a toggle for 'Enable Dashboard Creation' (on) and a note about allowing users to create new solid security dashboards. The 'Password Requirements' section contains a toggle for 'Strong Passwords' (on) and a note about requiring strong passwords.

Login Lockdown

This screenshot shows the 'Login Lockdown' settings page from the WP Login Lockdown plugin. The left sidebar is dark-themed and includes links for Dashboard, Posts, Media, Pages, Comments, Appearance, Plugins, Users, Limit Login Attempts, Tools, and Settings (which is highlighted). The main content area features a 'WP Login Lockdown' header with three status indicators: 'Lockdowns in last 24h' (0), 'Lockdowns since plugin installed' (0), and 'Failed logins in last 24h' (0). Below this are tabs for Login Protection (selected), Activity, Country Blocking, 2FA, Captcha, Cloud Protection, and Temp Access. The 'Basic' tab is active. The 'Max Login Retries' section has a value of 4 and a note about triggering a lockdown. The 'Retry Time Period Restriction' section has a value of 2 minutes and a note about the time before a lockdown. The 'Lockout Length' section has a value of 60 minutes and a note about the duration a IP is locked out. The 'Log Failed Attempts With Non-existent Usernames' section has a toggle switch turned on. The 'Mask Login Errors' section has a toggle switch turned off. The 'Block Type' section has two options: 'Completely block website access' (radio button) and 'Only block access to the login page' (radio button, which is selected). A green shield icon is visible in the bottom right corner.

Updraft Plus Backup

Cyber Security Blog 0 + New LLAR Security 2 UpdraftPlus Howdy, cloud

To make your site as secure as possible, take a moment to optimize the Wordfence Web Application Firewall: [CLICK HERE TO CONFIGURE](#)

[DISMISS](#)

If you cannot complete the setup process, [click here for help.](#)

UpdraftPlus Backup/Restore

Welcome to UpdraftPlus! To make a backup, just press the Backup Now button. [To change any of the default settings of what is backed up, to configure scheduled backups, to send your backups to remote storage \(recommended\), and more, go to the settings tab.](#)

[UpdraftPlus.Com](#) | [Premium](#) | [News](#) | [Twitter](#) | [Support](#) | [Newsletter sign-up](#) | [Lead developer's homepage](#) | [FAQs](#) | [More plugins](#) - Version: 1.24.4

[Backup / Restore](#) [Migrate / Clone](#) [Settings](#) [Advanced Tools](#) [Premium / Extensions](#)

Next scheduled backups:

Files:	Database:	
Nothing currently scheduled	Nothing currently scheduled	Backup Now

Time now: Sat, July 27, 2024 23:27 [Add changed files \(incremental backup\) ...](#)

Last log message:

The backup succeeded and is now complete (Jul 27 23:27:39) [Download most recently modified log file](#)

[Existing backups](#)

Enabling Two factor authentication for admin

[DISMISS](#)

If you cannot complete the setup process, [click here for help.](#)

[Two-Factor Authentication](#) [Settings](#) [Learn more about Two-Factor Authentication](#)

Two-Factor Authentication

Two-Factor Authentication, or 2FA, significantly improves login security for your website. Wordfence 2FA works with a number of TOTP-based apps like Google Authenticator, FreeOTP, and Authy. For a full list of tested TOTP-based apps, [click here](#).

Editing User: cloud (you)

1. Scan Code or Enter Key

Scan the code below with your authenticator app to add this account. Some authenticator apps also allow you to type in the text version instead.



RWDBFOYZBVLKGN37LSQXGGH3FC63ON2Y

2. Enter Code from Authenticator App

Download Recovery Codes Optional

Use one of these 5 codes to log in if you lose access to your authenticator device. Codes are 16 characters long plus optional spaces. Each one may be used only once.

1abb be14 e385 512a
bb7b 10c6 b2b4 f95f
217b c396 e339 07f1
c58b b472 87c9 ac66
9eea d1ed c8eb 8454

[DOWNLOAD](#)

Enter the code from your authenticator app below to verify and activate two-factor authentication for this account.

For help on setting up an app, visit our help article. [ACTIVATE](#)

Limiting Login Attempts

The screenshot shows a WordPress login screen. At the top, there is a large blue 'W' logo. Below it, a red error box displays the message: "ERROR: Too many failed login attempts. Please try again in 1 minute." To the right of the error box, a message says "You are now logged out." The main form contains fields for "Username or Email Address" (containing "sdsdsdscdc") and "Password" (represented by a series of black dots). There are also "Remember Me" and "Log In" buttons. Below the form, links for "Lost your password?" and "← Go to Cyber Security Blog" are visible.

Limited Attempts

The screenshot shows a WordPress login screen. At the top, there is a large blue 'W' logo. Below it, a red error box displays the message: "ERROR: Incorrect username or password." Below the error message, it says "3 attempts remaining." The main form contains fields for "Username or Email Address" (containing "cloud") and "Password" (represented by a series of black dots). There are also "Remember Me" and "Log In" buttons. Below the form, links for "Lost your password?" and "← Go to Cyber Security Blog" are visible.

List of plugins

The screenshot shows the WordPress admin dashboard with the 'Plugins' menu selected. The main content area displays a list of installed plugins, specifically focusing on security-related ones. The list includes:

Plugin	Description	Automatic Updates
Akismet Anti-spam: Spam Protection Settings Deactivate	Used by millions, Akismet is quite possibly the best way in the world to protect your blog from spam. Your site is fully configured and being protected, even while you sleep. Version 5.3.3 By Automattic - Anti-spam Team View details	Enable auto-updates
Limit Login Attempts Reloaded Dashboard Settings Deactivate	Block excessive login attempts and protect your site against brute force attacks. Simple, yet powerful tools to improve site performance. Version 2.26.12 By Limit Login Attempts Reloaded View details	Enable auto-updates
Login Lockdown Get EXTRA login protection Settings Deactivate	Protect the login form by banning IPs after multiple failed login attempts. Version 2.11 By WebFactory Ltd View details Support	Enable auto-updates
Solid Security Basic Settings Deactivate	Shield your site from cyberattacks and prevent security vulnerabilities. The only security plugin you need for a solid foundation. Version 9.3.3 By SolidWP View details Get Support	Enable auto-updates
UpdraftPlus - Backup/Restore Premium / Pro Support Settings Deactivate Take Tour	Backup and restore: take backups locally, or backup to Amazon S3, Dropbox, Google Drive, Rackspace, (S)FTP, WebDAV & email, on automatic schedules. Version 1.24.4 By UpdraftPlus.Com , DavidAnderson View details	Enable auto-updates
Wordfence Security Upgrade To Premium Deactivate	Wordfence Security - Anti-virus, Firewall and Malware Scan Version 7.11.6 By Wordfence View details	Enable auto-updates
Plugin	Description	Automatic Updates