Apache Notes

Aidan Fetter
10/02/2023

1. What is Apache?

Apache is a web server that allows us to host content online (web applications, files, etc.) for others to utilize
- Uses HTTPd (HTTP Daemon): A software program that usually runs as a background process and acts as a service

2. Installation:

sudo apt install apache2
sudo service apache2 start //starts apache
sudo apache2ctl configtest //tests if the configuration is valid
sudo service apache2 reload //reloads apache
OR
sudo systemctl restart apache2

3. Create a simple web page:

cd /var/www/html
vim, nano, cat, etc. index.html
Add html code
Locate the address and display webpage
nmap scan (using -sV or -O)

4. Security (reload after each instance):

4.1. Hide Apache version and operating system information when an error occurs:

Modify “server signature” = OFF and “ServerTokens Prod” (/etc/apache2/conf-available/security.config)

4.2. Disable directory listing (if there is no index file, the entire content of the folder will be displayed):

- Example using confidential file

```
<Directory /var/www/>
        Options Indexes FollowSymLinks
        AllowOverride None
        Require all granted
</Directory>
```

- Either remove “Indexes” or change it to “-Indexes”

4. Security (reload after each instance):

4.3. Disable Unnecessary modules:

- View all modules- apache2ctl -M
- Enable/Disable modules (respectively): “sudo a2enmod moduleName” and “sudo a2dismod moduleName”
- Modules can be found here

4.4. Restrict access to files outside the directory

- Prevents from directory traversal attacks
- Same area as directory listing
- “Require all granted” or “Require all denied”

For this section, split members into two teams. Each team will have ~20-30 minutes to implement what is shown below. Only the header is shown when presenting and the answers (shown below the header) are shown once the time is up. This allows the team to work together quickly and efficiently to achieve the task at hand. Remember, the competition is meant to be stressful!

------ Team 1

4.5. Prevent DoS attacks

There are several ways to do this step:

1. Using a third party module (such as mod_evasive)

The below options can be used together for maximum security and are very important when hosting a web server:

2. KeepAliveTimeout (/etc/apache2/apache.config)
   - Time ≤ 5
3. MaxKeepAliveRequests (/etx/apache2/apache.config)
   - Requests ≤ 100
4. KeepAlive (Depending on how many users you expect)
5. TimeOut – Decreasing can help mitigate slowloris attacks by reducing the time an attacker can keep a connection open
   - TimeOut ≤ 75-100
6. LimitRequestBody (If your server has it)
7. Post_max_size and Upload_max_filesize general areas in /etc/php/8.1/apache2/php.ini
8. Memory_limit (/etc/php/8.1/apache2/php.ini)

----- Team 2

4.6. Add a firewall, allow the HTTP and HTTP service to run permanently, and add ports 80 and 443 to the firewall

The best way to do this is to add the “mod_security” module that allows you to monitor traffic in real time and prevents the web server from brute force attacks.

- Use the command “sudo apt install libapache2-mod-security2” to install it and then restart apache

You can also install firewalld to implement a firewall and add rules to it:

- “sudo apt install firewalld”

To allow firewalld to work with the Apache HTTP/HTTPS server:

- “sudo systemctl start firewalld.service
- “sudo firewall-cmd –add-service http --permanent”
- “sudo firewall-cmd –add-service https --permanent"
- “sudo restart firewalld.service”
- “sudo enable firewalld.service”

To add ports to the firewall:

- “sudo systemctl start firewalld”
- To check statud: “sudo systemctl status firewalld”
- Enable firewall: “sudo systemctl enable firewalld”
- To open a port (ex: 80): “sudo firewall-cmd –add-port=80/tcp --permanent”
    - “--permanent" sets the options permanently; without it, the change will only be part of the runtime configuration
- “sudo firewall-cmd –reload”

For those using httpd to host the Apache2 server:

- Prevent other services from activating httpd by masking it: “sudo systemctl mask httpd”
- Disable httpd: “sudo systemctl disable httpd”
- Stop the httpd service: “sudo systemctl stop httpd”
- “sudo firewall-cmd --permanent --add-port=80/tcp”
- “sudo firewall-cmd --reload
- Enable and start the service: “sudo systemctl enable –now httpd”

5. Extra Resources

https://www.redhat.com/sysadmin/apache-basics

https://www.twaino.com/en/blog/website-creation/apache-server-2/#Chapter_5_How_to_secure_the_Apache_Web_server

5. Extra Resources

https://webhostinggeeks.com/blog/8-easy-steps-to-safeguard-an-apache-web-server-and-prevent-ddos/

https://access.redhat.com/documentation/en-us/red_hat_enterprise_linux/8/html/deploying_different_types_of_servers/setting-apache-http-server_deploying-different-types-of-servers

https://firewalld.org/documentation/man-pages/firewall-cmd.html