Apache hardening manuál

Tento dokument opisuje základné kroky potrebné pre zvýšenie bezpečnosti vlastnej inštancie webového servera Apache.

1. Skrytie informácií o verzii služby Apache a OS

Skrytie informácií o konkrétnej inštancii služby Apache je potrebné najmä z dôvodu prevencie prípadného kybernetického útoku. Poskytnutie informácií o konkrétnej verzii služby uľahčuje nájdenie bezpečnostných nedostatkov a zraniteľností.

Pre systémy založené na distribúcii Debian otvoríme súbor:

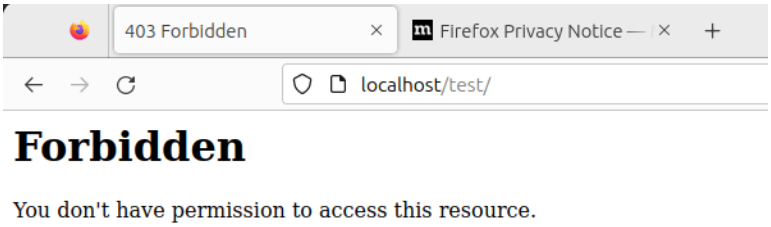
```
$ sudo vim /etc/apache2/apache2.conf
```

Do súboru pridáme nasledovné riadky:

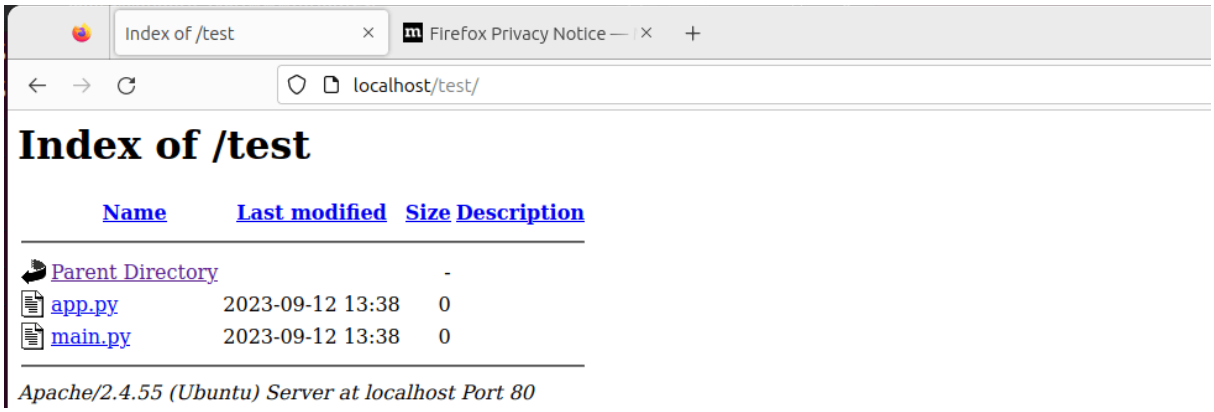
```
ServerTokens Prodx
ServerSignature Offx
```

Po úprave súboru reštartujeme službu (odporúčame vykonať mimo prevádzkových hodín):

```
$ sudo systemctl restart apache2
```



2. Vypnutie Directory listing-u



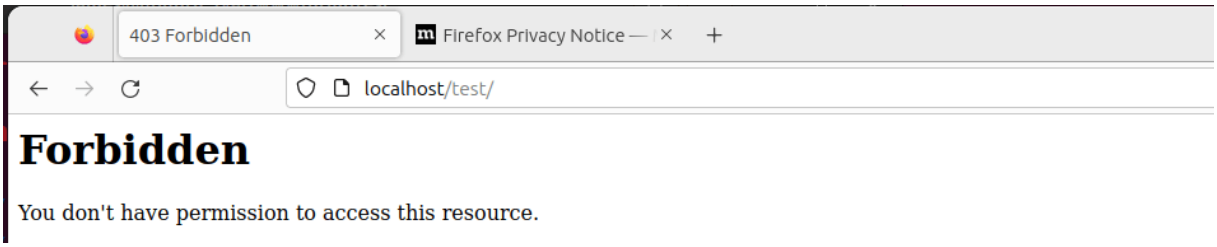
Obr. 1 Directory listing pred aplikovaním príkazu

Pre systémy založené na Debiane, otvoríme súbor:

```
$ sudo vim /etc/apache2/apache2.conf
```

Hodnotu atribútu *Directory* nastavíme na:

```
<Directory /opt/apache/htdocs>
Options -Indexes
</Directory>
```



Obr. 2 Directory listing po aplikovaní príkazu

Viac o directory listing-u nájdete na https://portswigger.net/kb/issues/00600100_directory-listing

3. Pravidelná aktualizácia služby Apache

Pre implementáciu bezpečnostných záplat vykonáme pravidelnú aktualizáciu pomocou príkazu `sudo apt update && sudo apt upgrade`, prípadne pomocou cronjob-u. Detailný manuál na vytvorenie cronjob-ov môžeme nájsť na https://www.hostinger.com/tutorials/cron-job.

4. Zabezpečenie pomocou SSL (použitie HTTPS)

Povolenie modulu SSL pre Apache:

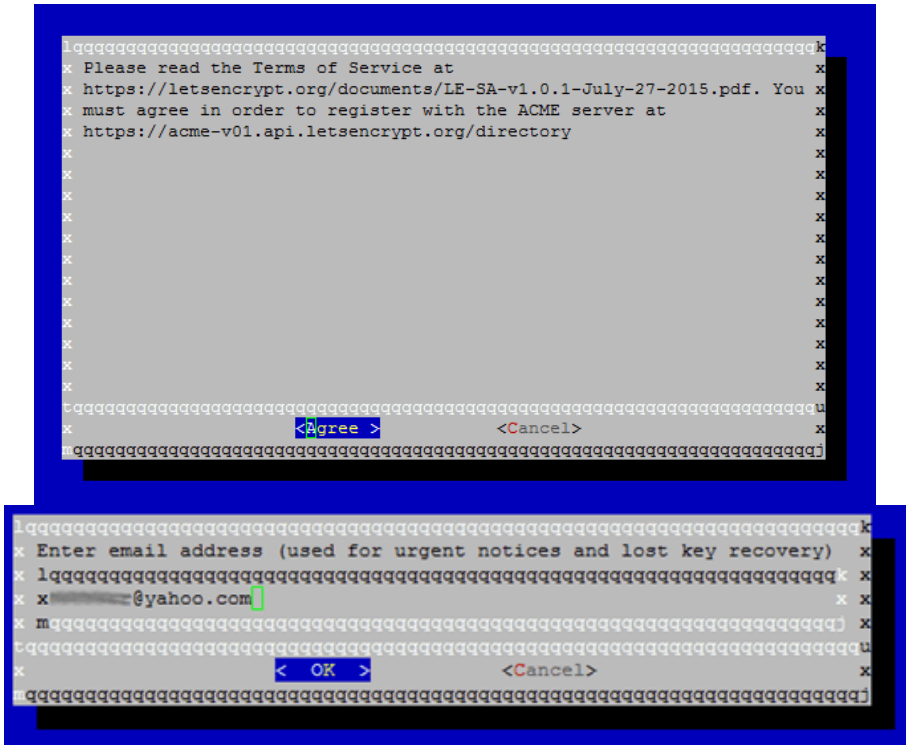
```
$ sudo a2enmod ssl
$ sudo a2ensite default-ssl.conf
$ sudo service apache2 restart
```

Stiahnutie šifrovacieho klienta:

```
$ sudo apt-get -y install git
$ cd /usr/local
$ sudo git clone https://github.com/letsencrypt/letsencrypt
```

Vygenerovanie bezpečnostného certifikátu službou Let's Encrypt (určeného na šifrovanie pomocou protokolu TLS)

```
$ cd /usr/local/letsencrypt
$ sudo ./letsencrypt-auto --apache -d your_domain.tld
```



Pre automatickú obnovu certifikátu po 90 dňoch vytvoríme cronjob:

```
$ sudo crontab –e
```

V Cron tabe pridáme nasledujúci záznam:

```
0 1 1 */2 * cd /usr/local/letsencrypt && ./letsencrypt-auto certonly --
apache --renew-by-default --apache -d domain.tld >> /var/log/domain.tld-
renew.log 2>&1
```

5. Povolenie HSTS (HTTP Strict Transport Security)

V nadväznosti na implementáciu SSL odporúčame zároveň povoliť HSTS, ktorá ochraňuje stránku voči útokom typu Man-In-The-Middle.

Povolenie modulu *headers:*

```
$ sudo a2enmod headers
```

Reštartovanie služby Apache:

```
$ sudo systemctl restart apache2
```

Úprava konfiguračného súboru:

```
$ sudo vim /etc/apache2/sites-available/mydomain.conf
```

Pridanie záznamu v bloku *<VirtualHost *:443>*:

```
Header always set Strict-Transport-Security "max-age=31536000;
includeSubDomains"
```

Opätovné reštartovanie služby Apache:
```
$ sudo systemctl restart apache2
```

6. Povolenie HTTP/2

HTTP/2 predstavuje novšiu a bezpečnejšiu verziu protokolu HTTP. Rozdiely medzi HTTP/2 a HTTP/1 môžeme nájsť na https://www.cloudflare.com/learning/performance/http2-vs-http1.1/.

Povolenie modulu HTTP/2:
```
$ sudo a2enmod http2
```

Úprava konfiguračného súboru SSL:
```
$ sudo vim /etc/apache2/sites-enabled/your-domain-name-le-ssl.conf
```

Pridanie záznamu v bloku *<VirtualHost *:443>*:
```
Protocols h2 http/1.1
```

Reštartovanie služby Apache:
```
$ sudo systemctl restart apache2
```

7. Zamedzenie prístupu k citlivým súborom

V bloku *<VirtualHost *:80>*: súboru */etc/apache2/sites-enabled/example_name.conf* pomocou záznamu *directory* a *Require all denied* zakážeme prístup k priečinku pre všetkých používateľov.

```
<VirtualHost *:80>
    ServerName example.com
    DocumentRoot /var/www/html

    # Other virtual host settings

    <Directory /var/www/html/sensitive_directory>
        Require all denied
    </Directory>
</VirtualHost>
```

Reštartovanie služby Apache:
```
$ sudo systemctl restart apache2
```

8. Vypnutie päty ServerSignature Directive

ServerSignature Directive vytvára pätu schránky, do ktorej vkladá informácie o konfigurácii servera Apache. Rovnako ako pre Skrytie informácií o verzii služby Apache a OS, aj tu platí, že zverejnenie prílišného množstva informácií má za následok zvýšenie rizika kompromitácie. Pre jej odstránenie je potrebné vykonať nasledovné:

Vloženie záznamu do konfiguračného súboru Apache:
```
ServerSignature Off
```

Reštartovanie služby Apache:
```
$ sudo systemctl restart apache2
```

9. Nastavenie atribútu ServerTokens na Prod

ServerTokens určuje, aké informácie o službe Apache aplikácia odosiela. Nastavením hodnoty na *prod* povolíme odosielanie len najnutnejších informácií.

Vloženie záznamu do konfiguračného súboru Apache:
```
ServerTokens prod
```

Reštartovanie služby Apache:
```
$ sudo systemctl restart apache2
```

10. Vypnutie nepotrebných modulov

Moduly predstavujú externé programy a funkcie využívané v rámci služby Apache. Vypnutím nepotrebných modulov tak zmenšíme možnosti kompromitácie systému na nevyhnutné minimum.

Zobrazenie všetkých povolených modulov:
```
$ apache2ctl –M
```

Vypnutie ľubovoľného modulu:
```
$ sudo a2dismod <modul>
```

11. Povolenie logovania

Logovanie nám poskytuje detailný pohľad na udalosti, ktoré sa dejú a udiali v systéme. Tieto informácie môžu byť veľmi dôležité v procese riešenia bezpečnostného incidentu.

Pre povolenie logovania je nutné importovať modul *mod_log_config.*
V rámci súboru VirtualHost následne vložíme atribúty *ErrorLog* a *CustomLog.*
```
<VirtualHost x.x.x.x:x>
        ServerName example.com
        DocumentRoot /var/www/html/example/
        ErrorLog ${APACHE_LOG_DIR}/error.log
        CustomLog ${APACHE_LOG_DIR}/access.log combined
</VirtualHost>
```

12. Používanie Apache so samostatným používateľom a skupinou

Pomocou tohto nastavenia izolujeme Apache od ostatných procesov.

Vytvorenie skupiny, používateľa a jeho priradenie do skupiny:
```
$ sudo groupadd apachegroup
$ sudo useradd -g apachegroup apacheuser
```

Zmena záznamov *user* a *group* v konfiguračnom súbore Apache:
```
User apacheuser
Group apachegroup
```

Vzhľadom na zmenu používateľa pre Apache je zároveň potrebné zmeniť majiteľa všetkých priečinkov a súborov, ku ktorým má mať Apache prístup:
```
$ sudo chown -R apacheuser:apachegroup /path/to/dir_or_file
```

Reštartovanie služby Apache:
```
$ sudo systemctl restart apache2
```

13. Ochrana binárnych a konfiguračných priečinkov

Základné povolenia pre tieto súbory sú 755, čo znamená, že každý používateľ si môže dané súbory prezerať.

Zmena povolení pre daný súbor:
```
$ chmod –R 750 dir
```

Viac o nastavení povolení na https://www.geeksforgeeks.org/permissions-in-linux/.

14. Nastavenie hlavičiek HttpOnly a Secure, X-Frame-Options, X-XSS-Protection

Pridaním týchto hlavičiek znížime pravdepodobnosť manipulácie s tzv. cookies a tým pádom aj útokov typu Cross Site Scripting. X-Frame-Options zabraňuje útokom typu Clickjacking, kedy obeť neúmyselne klikne na objekt, ktorý nie je vidieť. X-XSS-Protection zabraňuje načítaniu stránok v prípade detekcie XSS útoku.

Vloženie záznamu do konfiguračného súboru Apache:
```
Header edit Set-Cookie ^(.*)$ $1;HttpOnly;Secure
Header always append X-Frame-Options SAMEORIGIN
Header set X-XSS-Protection "1; mode=block"
```

Reštartovanie služby Apache:
```
$ sudo systemctl restart apache2
```

Vhodným doplnkom je zároveň pridanie hlavičky HSTS. Táto vynucuje používanie šifrovaného pripojenia na webstránku. Viac o nastavení hlavičky HSTS na https://developer.mozilla.org/en-US/docs/Web/HTTP/Headers/Strict-Transport-Security.

15. Vykonávanie pravidelných skenov

V rámci dôkladného zabezpečenia služby Apache je vhodné vykonávať pravidelné skeny zraniteľností a aplikovať bezpečnostné záplaty. Medzi obľúbené nástroje patria napr. Acutenix, Nessus, Nexpose, Sucuri a iné.

Takúto službu poskytuje aj VJ CSIRT. Pre registráciu navštívte https://www.csirt.gov.sk/registracia-achilles.html?csrt=6580922874488254178.

16. Limitácia povolení pre prístupové metódy

Vloženie nasledujúceho kódu do konfiguračného súboru */etc/apache2/apache2.conf*
```
<Directory "url/to/restrict">
    <LimitExcept GET POST>
        Deny from all
    </LimitExcept>
</Directory>
```

Reštartovanie služby Apache:
```
$ sudo systemctl restart apache2
```

17. Zálohovanie servera

Záloha servera je jednou z najdôležitejších častí spravovania služieb. Zálohu je možné použiť pre obnovu po kompromitácii, nechcenej zmene a pod.

Zálohu môžeme vykonať príkazom

```
sudo cp /etc/apache2/apache2.conf /etc/apache2/apache2.conf.bak
```

18. Implementácia firewall-u pre zabezpečenie služby

Dobrou praxou pre zabezpečenie našich serverov je využitie FW, napr. ufw, iptables, a pod. V tomto prípade ide najmä o zabezpečenie podporných služieb tak, aby sme zachovali potrebnú funkcionalitu, ale zároveň zabránili neoprávnenému prístupu a úpravám.
Detailnejší návod nájdeme na https://www.ugacomp.com/how-to-configure-iptables-to-secure-apache-server-on-ubuntu/.

Zdroje

https://www.tecmint.com/apache-security-tips/
https://geekflare.com/apache-web-server-hardening-security/
https://httpd.apache.org/docs/2.4/misc/security_tips.html
https://www.tutorialspoint.com/10-apache-web-server-security-and-hardening-tips
https://cheatsheetseries.owasp.org/cheatsheets/HTTP_Headers_Cheat_Sheet.html#introduction.