Apache **Hardening Manuál**

Praktický postup zabezpečenia webového servera Apache.

OBSAH

Úvod

Tento dokument opisuje základné kroky potrebné pre zvýšenie bezpečnosti vlastnej inštancie webového servera Apache. Postup je navrhnutý tak, aby na jeho konci vznikla funkčná, bezpečná konfigurácia vhodná pre produkčné prostredie.

Kroky sú zoradené podľa poradia, v akom ich odporúčame vykonávať. Na konci dokumentu nájdete prehľad zdrojov a odkazov na ďalšie informácie.

Tento návod predpokladá prístup ku konfiguračným súborom servera na VPS alebo dedikovanom serveri s distribúciou založenou na Debiane. Ak používate zdieľaný hosting, obráťte sa na svojho poskytovateľa.

Pred začatím

⚠️ **PRED AKÝMIKOĽVEK ZMENAMI SI VYTVORTE ZÁLOHU**

Vždy zálohujte existujúce konfiguračné súbory pred ich úpravou. Zálohu konfigurácie Apache vykonáte príkazom:

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

Funkčnú konfiguráciu si uložte na bezpečné miesto mimo servera. Ak dôjde k problému, záloha je vaša jediná možnosť rýchlej obnovy.

Čo budete potrebovať

- Prístup k serveru
- Základnú znalosť práce s príkazovým riadkom
- Textový editor na serveri (napr. `vim` alebo `nano`)

✅ **OTESTUJTE KONFIGURÁCIU PRED KAŽDÝM REŠTARTOM**

Po každej zmene konfiguračného súboru spustite príkaz `sudo apache2ctl configtest`. Ak príkaz hlási chybu, Apache konfiguráciu nenačíta. Odporúčame reštartovanie služby vykonávať mimo prevádzkových hodín.

01

KAPITOLA PRVÁ

Základná konfigurácia

Kroky, ktoré vykonáte hneď po inštalácii. Výsledkom je server so skrytými informáciami, správnymi oprávneniami a bez zbytočne otvorených možností útoku.

1 Skrytie informácií o verzii a OS

Predvolene Apache v HTTP odpovediach zverejňuje svoju verziu a informácie o operačnom systéme. Útočníci tieto informácie využívajú na vyhľadávanie známych zraniteľností pre konkrétnu verziu. Zverejnenie je potrebné vypnúť.

Ukážka zverejnenia informácií o verzií systému Apache.

Otvorte súbor `/etc/apache2/apache2.conf` :

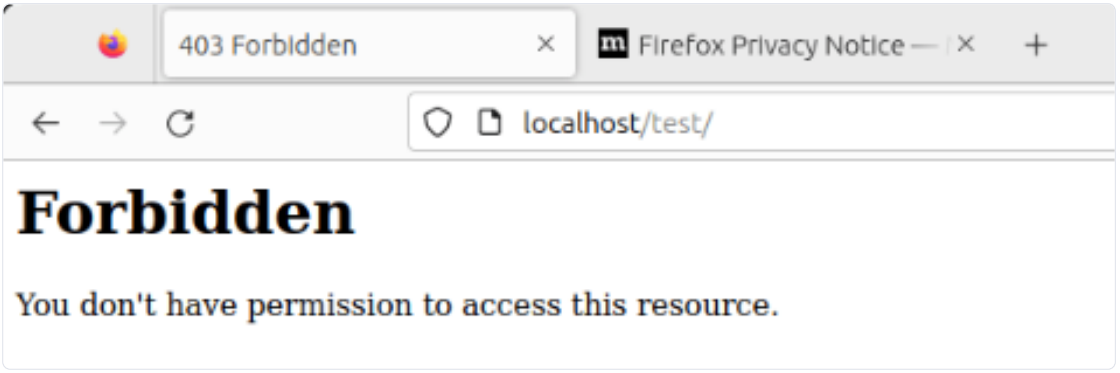
```
sudo vim /etc/apache2/apache2.conf
```

Do súboru pridajte nasledovné riadky:

```
ServerTokens Prod
ServerSignature Off
```

Po úprave reštartujte službu:

```
sudo systemctl restart apache2
```

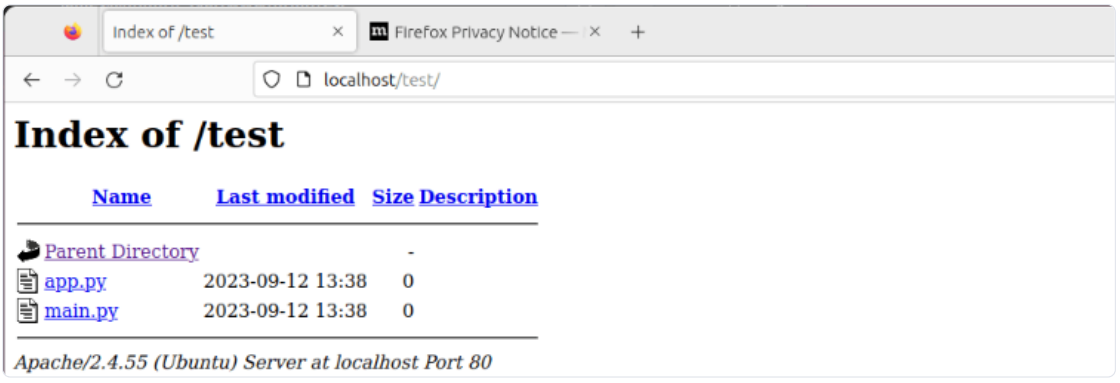


Ukážka správneho nastavenia bez zverejnenia informácií o verzii systému Apache.

> 📋 **ČO TIETO DIREKTÍVY ROBIA**
>
> `ServerTokens Prod` obmedzí hlavičku `Server:` len na reťazec `Apache` bez verzie a OS.
> `ServerSignature Off` odstráni pätu so serverovými informáciami, ktorú Apache pridáva na automaticky generované stránky (napr. chybové stránky 404).

2 Vypnutie výpisu adresárov

Ak adresár neobsahuje indexový súbor, Apache môže zobraziť zoznam jeho obsahu. Táto funkcia prezrádza štruktúru servera a musí byť vypnutá.



Ukážka nesprávneho nastavenia Apache, ktorý umožňuje voľné listovanie adresárov

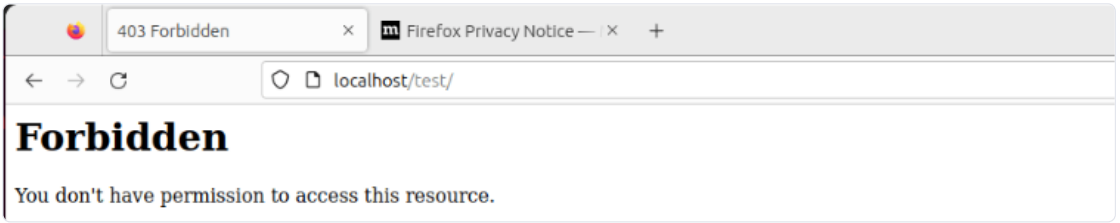
Otvorte súbor `/etc/apache2/apache2.conf` a pre príslušný adresár nastavte:

```
<Directory /var/www/html>
    Options -Indexes
</Directory>
```

Reštartujte službu:

```
sudo systemctl restart apache2
```

> ✅ **OVERENIE NASTAVENIA**
>
> Vytvorte prázdny testovací adresár bez indexového súboru a skúste ho načítať v prehliadači alebo cez `curl`. Mali by ste dostať odpoveď `403 Forbidden`, nie zoznam súborov.



Listovanie adresárov po aplikovaní odporúčaných nastavení

3 Bežná prevádzka pod samostatným používateľom

Apache worker procesy by mali byť spustené pod samostatným, izolovaným používateľom s minimálnymi oprávneniami. Tým oddelíme Apache od ostatných systémových procesov.

Vytvorte skupinu a používateľa:

```
sudo groupadd apachegroup
sudo useradd -g apachegroup apacheuser
```

Upravte záznamy `User` a `Group` v konfiguračnom súbore Apache:

```
User apacheuser
Group apachegroup
```

Zmeňte vlastníka všetkých adresárov a súborov, ku ktorým má mať Apache prístup:

```
sudo chown -R apacheuser:apachegroup /path/to/dir_or_file
```

Reštartujte službu:

```
sudo systemctl restart apache2
```

4 Ochrana konfiguračných adresárov

Predvolené oprávnenia pre konfiguračné a binárne súbory Apache sú `755`, čo znamená, že každý používateľ systému si môže dané súbory prezerať. Toto je potrebné obmedziť.

Nastavte oprávnenia `750` pre príslušné adresáre:

```
chmod -R 750 /etc/apache2
```

> 📋 **OPRÁVNENIA V LINUXE**
>
> Oprávnenie `750` znamená: vlastník má plný prístup (čítanie, zápis, spustenie), skupina má prístup na čítanie a spustenie, ostatní používatelia nemajú žiadny prístup. Viac o nastavení oprávnení na **geeksforgeeks.org/permissions-in-linux**.

5 Vypnutie nepotrebných modulov

Moduly predstavujú externé programy a funkcie využívané v rámci Apache. Každý aktívny modul rozširuje plochu možného útoku. Vypnutím nepotrebných modulov túto plochu zmenšíme na nevyhnutné minimum.

Zobrazte všetky povolené moduly:

```
apache2ctl -M
```

Vypnite ľubovoľný nepotrebný modul:

```
sudo a2dismod <modul>
```

Reštartujte službu:

```
sudo systemctl restart apache2
```

> ⚠️ **OVERTE KOMPATIBILITU PRED VYPNUTÍM MODULU**
>
> Pred vypnutím každého modulu overte, či ho vaša aplikácia nevyžaduje. Vypnutie nesprávneho modulu môže spôsobiť nefunkčnosť aplikácie alebo samotného Apache.

02

KAPITOLA DRUHÁ

SSL/TLS a HTTPS

Šifrovanie komunikácie medzi serverom a návštevníkmi je dnes nevyhnutnosťou. Táto kapitola vás prevedie povolením SSL, získaním certifikátu a konfiguráciou moderných protokolov.

6 Povolenie SSL a získanie certifikátu

Povolte modul SSL a predvolený SSL virtual host:

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

Inštalácia Let's Encrypt klienta

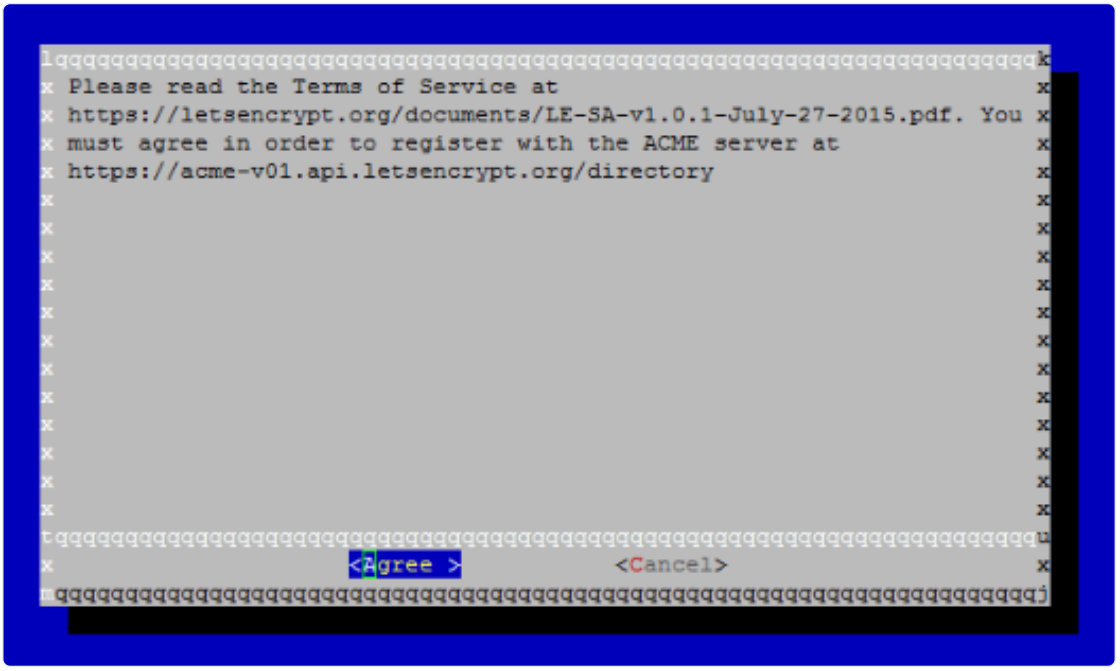
Na získanie a správu bezplatných SSL/TLS certifikátov použijeme nástroj Let's Encrypt:

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

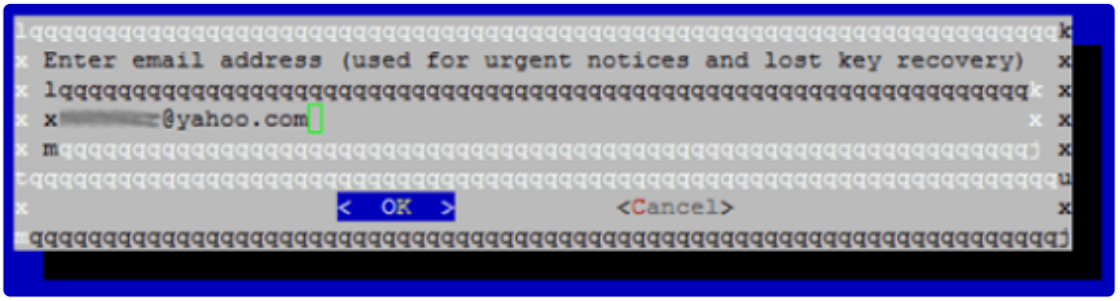
Vygenerovanie certifikátu

Nahraďte `your_domain.tld` názvom vašej domény:

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



Proces generovania certifikátu Let's Encrypt



Proces generovania certifikátu Let's Encrypt

Automatická obnova certifikátu

Certifikáty Let's Encrypt sú platné 90 dní. Pre automatickú obnovu vytvorte cronjob:

```
sudo crontab -e
```

Do crontabu pridajte nasledujúci záznam, ktorý spustí obnovu každé 2 mesiace:

```
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
```

✅ **OVERENIE PLATNOSTI CERTIFIKÁTU**

Detailný manuál na vytvorenie a správu cronjobov nájdete na **hostinger.com/tutorials/cron-job**. Platnosť certifikátu môžete kedykoľvek skontrolovať online nástrojom SSL Labs na adrese **ssllabs.com/ssltest**.

7 Povolenie HSTS

HTTP Strict Transport Security (HSTS) inštruuje prehliadač, aby sa na vašu stránku vždy pripájal výhradne cez HTTPS. Ochraňuje stránku voči útokom typu Man-In-The-Middle.

Povolte modul headers a reštartujte Apache:

```
sudo a2enmod headers
sudo systemctl restart apache2
```

Otvorte konfiguračný súbor virtual hostu:

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

Do bloku `<VirtualHost *:443>` pridajte:

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

Reštartujte službu:

```
sudo systemctl restart apache2
```

> ⚠️ **HSTS JE ŤAŽKO ZVRATITEĽNÉ**
>
> Po aktivácii HSTS prehliadače odmietnu načítať stránku cez HTTP počas celej doby platnosti definovanej v `max-age` (tu 1 rok). Pred nasadením sa uistite, že HTTPS funguje spoľahlivo na celej doméne vrátane subdomén. Ak si nie ste istí, začnite s hodnotou `max-age=300` (5 minút) a po overení funkčnosti ju zvýšte.

8 Povolenie HTTP/2

HTTP/2 je novšia a výkonnejšia verzia protokolu HTTP s lepšou podporou bezpečnostných funkcií. Povolte modul HTTP/2:

```
sudo a2enmod http2
```

Otvorte konfiguračný súbor SSL virtual hostu:

```
sudo vim /etc/apache2/sites-enabled/your-domain-name-le-ssl.conf
```

Do bloku `<VirtualHost *:443>` pridajte:

```
Protocols h2 http/1.1
```

Reštartujte službu:

```
sudo systemctl restart apache2
```

> 📋 **ROZDIELY MEDZI HTTP/1 A HTTP/2**
>
> HTTP/2 prináša multiplexovanie požiadaviek, kompresiu hlavičiek a server push, čo výrazne zlepšuje výkon. Prehľad rozdielov nájdete na **cloudflare.com/learning/performance/http2-vs-http1.1**.

03

KAPITOLA TRETIA

Bezpečnostné HTTP hlavičky

Bezpečnostné hlavičky informujú prehliadač, ako má narábať s obsahom vašej stránky. Správne nastavené hlavičky výrazne znižujú riziko útokov ako XSS, clickjacking alebo manipulácia s cookies.

9 Základné bezpečnostné hlavičky

Uistite sa, že modul `headers` je povolený (ak ste tak neurobili v kapitole 2):

```
sudo a2enmod headers
sudo systemctl restart apache2
```

Pridajte nasledujúce hlavičky do konfiguračného súboru Apache alebo do príslušného bloku `<VirtualHost>`:

```
Header edit Set-Cookie ^(.*)$ $1;HttpOnly;Secure
Header always append X-Frame-Options SAMEORIGIN
Header set X-Content-Type-Options "nosniff"
Header always set Referrer-Policy "strict-origin-when-cross-origin"
Header always set Permissions-Policy "geolocation=(), microphone=(), camera=(), payment=()"
```

Reštartujte službu:

```
sudo systemctl restart apache2
```

Hlavička	Účel
`Set-Cookie: HttpOnly; Secure`	Zabraňuje prístupu JavaScriptu ku cookies a vyžaduje HTTPS pre ich prenos. Ochrana pred XSS a session hijackingom
`X-Frame-Options: SAMEORIGIN`	Zabraňuje načítaniu stránky v iframe z cudzích domén. Ochrana pred clickjackingom
`X-Content-Type-Options: nosniff`	Zakazuje prehliadaču hádať typ súboru, ochrana pred MIME sniffingom
`Referrer-Policy`	Obmedzuje, ktoré informácie sa posielajú pri prechode na iný web
`Permissions-Policy`	Zakazuje prístup k hardvéru (kamera, mikrofón, GPS, platby)

⚠ **HLAVIČKU X-XSS-PROTECTION NEPOUŽÍVAJTE**

Aj keď pôvodný dokument túto hlavičku obsahoval, dnes je považovaná za zastaranú. Moderné prehliadače ju nepodporujú a v niektorých prípadoch môže sama o sebe zraniteľnosti spôsobiť. Vynechajte ju úplne, alebo jej priraďte hodnotu `0`. Ochranu pred XSS zabezpečuje správne nastavená Content Security Policy (sekcia 10).

10 Content Security Policy

Content Security Policy (CSP) je najúčinnejšia ochrana pred útokmi XSS. Definuje, z akých zdrojov môže prehliadač načítať skripty, štýly a ďalší obsah.

Pridajte do bloku `<VirtualHost *:443>`:

```
Header always set Content-Security-Policy \
  "default-src 'self'; script-src 'self' 'unsafe-inline'; \
  style-src 'self' 'unsafe-inline'; img-src 'self' data: https: blob:; \
  connect-src 'self' https:; font-src 'self' data:;"
```

⚠ **OTESTUJTE CSP PRED OSTRÝM NASADENÍM**

Nesprávne nastavená CSP môže narušiť funkčnosť stránky. Pred ostrým nasadením použite hlavičku `Content-Security-Policy-Report-Only`, ktorá len zaznamenáva porušenia do konzoly prehliadača (F12 → Console) bez blokovania. Po overení funkčnosti ju nahraďte ostrou verziou.

11 Overenie bezpečnostných hlavičiek

Po nasadení hlavičiek overte ich správnosť. Najprv skontrolujte, že Apache hlavičky skutočne odosiela:

```
curl -I https://vasa-stranka.sk
```

V odpovedi by ste mali vidieť všetky pridané hlavičky. Potom overte celkové hodnotenie online:

1. Navštívte **securityheaders.com** alebo **observatory.mozilla.org**
2. Zadajte adresu vašej domény
3. Cieľom je hodnotenie **A** alebo **A+**

04

KAPITOLA ŠTVRTÁ

Obmedzenie prístupu

Nastavenia, ktoré určujú, kto a čo má prístup k vášmu serveru. Patrí sem blokovanie citlivých adresárov, obmedzenie HTTP metód a konfigurácia firewallu.

12 Zamedzenie prístupu k citlivým adresárom

Pomocou direktívy `Require all denied` zakážeme prístup k citlivým adresárom pre všetkých používateľov.

Otvorte konfiguračný súbor virtual hostu:

```
sudo vim /etc/apache2/sites-enabled/example_name.conf
```

Do bloku `<VirtualHost *:80>` (alebo `*:443`) pridajte:

```
<VirtualHost *:443>
    ServerName example.com
    DocumentRoot /var/www/html

    # Blokovanie citlivého adresára
    <Directory /var/www/html/sensitive_directory>
        Require all denied
    </Directory>
</VirtualHost>
```

Reštartujte službu:

```
sudo systemctl restart apache2
```

✅ **OVERENIE BLOKOVANIA**

Po reloade otestujte blokovanie citlivých adresárov:

```
curl -I https://vasa-stranka.sk/sensitive_directory/
```

Odpoveď musí byť `403 Forbidden`. Ak dostanete `200 OK`, skontrolujte konfiguráciu bloku `Directory`.

13 Obmedzenie povolených HTTP metód

Pre bežnú webovú stránku sú potrebné len metódy `GET` a `POST`. Ostatné metódy je potrebné zakázať.

Do konfiguračného súboru `/etc/apache2/apache2.conf` pridajte:

```
<Directory "/var/www/html">
    <LimitExcept GET POST>
        Deny from all
    </LimitExcept>
</Directory>
```

Reštartujte službu:

```
sudo systemctl restart apache2
```

⚠️ **OVERTE KOMPATIBILITU S VAŠOU APLIKÁCIOU**

Niektoré webové aplikácie, REST API alebo pluginy využívajú metódy `PUT` alebo `DELETE`. Pred nasadením toto nastavenie otestujte a v prípade potreby pridajte ďalšie metódy do zoznamu za `LimitExcept`.

14 Konfigurácia firewallu

Firewall určuje, ktoré porty a služby sú dostupné z internetu. Pre väčšinu webových serverov stačí povoliť len tri porty: SSH, HTTP a HTTPS.

⚠️ **NAJPRV POVOĽTE SSH, AŽ POTOM ZAPNITE FIREWALL**

Ak zapnete firewall bez povolenia SSH portu, stratíte prístup k serveru. Poradie krokov v tejto sekcii je kritické, dodržte ho presne.

Nainštalujte a nakonfigurujte `ufw`:

```
sudo apt install ufw -y

# Najprv SSH - bez tohto sa po zapnutí firewallu zamknete von
sudo ufw allow 22/tcp

# HTTP a HTTPS pre webový server
sudo ufw allow 80/tcp
sudo ufw allow 443/tcp

# Až potom zapnite firewall
sudo ufw enable
```

Potvrďte aktiváciu stlačením **y**. Overte stav:

```
sudo ufw status verbose
```

Výstup by mal zobrazovať `Status: active` a tri povolené pravidlá pre porty 22, 80 a 443. Podrobnejší návod na konfiguráciu `iptables` nájdete na **ugacomp.com/how-to-configure-iptables-to-secure-apache-server-on-ubuntu**.

05

KAPITOLA PIATA

Logovanie a priebežná údržba

Bezpečnosť nie je jednorazový úkon. Logovanie poskytuje prehľad o udalostiach na serveri a pravidelná údržba udržuje váš server v bezpečnom stave.

15 Konfigurácia logovania

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

Pre povolenie logovania importujte modul `mod_log_config` a v rámci súboru VirtualHost vložte atribúty `ErrorLog` a `CustomLog`:

```
<VirtualHost x.x.x.x:443>
    ServerName example.com
    DocumentRoot /var/www/html/example/
    ErrorLog ${APACHE_LOG_DIR}/error.log
    CustomLog ${APACHE_LOG_DIR}/access.log combined
</VirtualHost>
```

Umiestnenie logov

Apache predvolene zapisuje logy do:

- `/var/log/apache2/access.log` - každá požiadavka na server
- `/var/log/apache2/error.log` - chyby a varovania

```
sudo tail -f /var/log/apache2/access.log
sudo tail -f /var/log/apache2/error.log
```

16 Pravidelná aktualizácia Apache

Zastaraná verzia Apache môže obsahovať bezpečnostné zraniteľnosti. Aktualizujte pravidelne v rámci celkového procesu aktualizácie systému:

```
sudo apt update && sudo apt upgrade -y
```

✅ **AUTOMATICKÉ AKTUALIZÁCIE**

Pre automatickú aplikáciu bezpečnostných záplat zvážte nasadenie nástroja **unattended-upgrades**. Prípadne nastavte automatickú aktualizáciu pomocou cronjob-u. Detailný manuál na vytvorenie cronjobov nájdete na **hostinger.com/tutorials/cron-job**.

17 Zálohovanie konfigurácie

Konfiguračné súbory sú výsledkom vášho úsilia. Zálohujte ich pred každou väčšou zmenou aj pravidelne.

```
# Záloha hlavného konfiguračného súboru Apache
sudo cp /etc/apache2/apache2.conf /etc/apache2/apache2.conf.bak

# Záloha celého konfiguračného adresára s dátumom
sudo cp -r /etc/apache2 /etc/apache2.bak.$(date +%Y%m%d)
```

18 Pravidelné skenovanie zraniteľností

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

✅ **SKENOVANIE ZRANITEĽNOSTÍ CEZ VJ CSIRT**

Túto službu poskytuje aj VJ CSIRT. Pre registráciu navštívte **csirt.sk**.

Čo ďalej

Dokončením tohto manuálu máte funkčný Apache server so bezpečnou základnou konfiguráciou. Bezpečnosť je však kontinuálny proces. Nasledujúce kroky vám pomôžu udržať a ďalej posilniť ochranu vašej infraštruktúry.

- **Automatické aktualizácie systému** - nasaďte nástroj *unattended-upgrades*, ktorý je podrobne opísaný v samostatnom manuáli **csirt.sk/unattended-upgrades**
- **Pravidelný audit** - mesačne overujte bezpečnostné hlavičky na **securityheaders.com** a SSL konfiguráciu na **ssllabs.com/ssltest**.
- **Pravidelné skenovanie** - VJ CSIRT poskytuje službu skenovania zraniteľností pre registrované organizácie na adrese **csirt.sk**.

Zdroje

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