NGINX **Hardening Manuál**

Praktický postup zabezpečenia webového servera NGINX.

OBSAH

Úvod

Tento dokument opisuje základné kroky potrebné pre zvýšenie bezpečnosti webového servera NGINX. 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 vzorové konfiguračné súbory ako rýchlu referenciu.

Tento návod predpokladá prístup ku konfiguračným súborom servera na VPS alebo dedikovanom serveri. Ak používate zdieľaný hosting, obráťte sa na svojho poskytovateľa. Väčšina krokov z Kapitoly 2 (SSL/TLS) a Kapitoly 3 (HTTP hlavičky) môže byť dostupná cez ovládací panel hostingu.

Pred začatím

⚠️ **PRED AKÝMIKOĽVEK ZMENAMI SI VYTVORTE ZÁLOHU**

Vždy zálohujte existujúce konfiguračné súbory pred ich úpravou. 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. `nano`)

✅ **OTESTUJTE KONFIGURÁCIU PRED KAŽDÝM RELOADOM**

Po každej zmene konfiguračného súboru spustite príkaz `sudo nginx -t`. Ak príkaz hlási chybu, NGINX konfiguráciu nenačíta a server zostane funkčný s predchádzajúcim nastavením. Nikdy nereštartujte NGINX bez úspešného testu.

Dva hlavné konfiguračné súbory

Súbor	Účel
`/etc/nginx/nginx.conf`	Hlavná konfigurácia. Globálne nastavenia platné pre celý server (timeouty, rate limiting zóny, logovanie)
`/etc/nginx/sites-available/vasa-stranka`	Konfigurácia konkrétnej stránky (virtual host). SSL, hlavičky, blokovanie súborov, location bloky

Každá sekcia tohto návodu jasne označuje, do ktorého súboru patrí daná zmena.

01

KAPITOLA PRVÁ

Inštalácia a základná konfigurácia

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

1 Inštalácia NGINX

Na systémoch Ubuntu a Debian nainštalujte NGINX z oficiálnych repozitárov:

```
sudo apt update
sudo apt install nginx -y
```

Po inštalácii spustite službu a nastavte jej automatické spúšťanie po reštarte systému:

```
sudo systemctl start nginx
sudo systemctl enable nginx
```

Overte, že NGINX je spustený:

```
sudo systemctl status nginx
```

Výstup by mal obsahovať riadok `Active: active (running)`. Ak sa vo vašom prehliadači po zadaní IP adresy servera zobrazí predvolená stránka NGINX, inštalácia prebehla úspešne.

2 Skrytie verzie servera

Predvolene NGINX v HTTP odpovediach zverejňuje svoju verziu. Útočníci tieto informácie využívajú na vyhľadávanie známych zraniteľností pre konkrétnu verziu. Zverejnenie je potrebné vypnúť.

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

```
sudo nano /etc/nginx/nginx.conf
```

Nájdite existujúci blok `http { ... }` a pridajte alebo overte prítomnosť tohto riadku:

```
http {
    server_tokens off;
    # ... zvyšok konfigurácie
}
```

Uložte súbor, otestujte konfiguráciu a načítajte ju:

```
sudo nginx -t
sudo systemctl reload nginx
```

> ✅ **OVERENIE**
>
> Spustite príkaz `curl -I https://vasa-stranka.sk` . V hlavičke `Server:` by ste mali vidieť len `nginx` bez čísla verzie.

3 Bežná prevádzka pod neprivilegovaným používateľom

NGINX worker procesy by mali byť spustené pod samostatným, izolovaným používateľom s minimálnymi oprávneniami. Na Ubuntu a Debian to zabezpečuje predvolený používateľ `www-data` .

Overte nastavenia v súbore `/etc/nginx/nginx.conf` na začiatku súboru, pred blokom `http { ... }`:

```
user www-data;
worker_processes auto;
```

Hodnota `auto` pre `worker_processes` automaticky nastaví počet workerov podľa počtu jadier procesora, nie je potrebné nič meniť.

4 Vypnutie voľného listovania adresárov

Ak adresár neobsahuje indexový súbor, NGINX môže zobraziť zoznam jeho obsahu.

Otvorte súbor `/etc/nginx/sites-available/vasa-stranka` a v bloku `location / { ... }` nastavte:

```
location / {
    autoindex off;
    try_files $uri $uri/ =404;
    # ... zvyšok konfigurácie location bloku
}
```

Nastavenie `try_files` hovorí NGINX-u: skús nájsť súbor, potom adresár, inak vráť 404. Bez tohto nastavenia sa môže NGINX pri nenájdenom súbore správať nepredvídateľne.

Vlastná stránka 404

Predvolená 404 stránka NGINX prezrádza názov servera. Vytvorte vlastnú stránku a zaregistrujte ju v bloku `server { ... }` v súbore `/etc/nginx/sites-available/vasa-stranka`:

```
error_page 404 /404.html;
location = /404.html {
    internal;
}
```

Súbor `404.html` uložte do koreňového adresára vašej stránky.

✅ **OVERENIE NASTAVENIA**

Vytvorte prázdny testovací adresár a skúste ho načítať:

```
sudo mkdir /var/www/vasa-stranka/test
curl -I https://vasa-stranka.sk/test/
```

Mali by ste dostať odpoveď `403 Forbidden` alebo `404 Not Found`, nie zoznam súborov. Po overení adresár zmažte: `sudo rm -r /var/www/vasa-stranka/test`

5 Obmedzenie povolených HTTP metód

Pre bežnú webovú stránku sú potrebné len metódy `GET`, `POST` a `HEAD`.

V súbore `/etc/nginx/sites-available/vasa-stranka`, v bloku `location / { ... }`, pridajte:

```
location / {
    limit_except GET POST HEAD {
        deny all;
    }
    # ... zvyšok konfigurácie location bloku
}
```

⚠️ **OVERTE KOMPATIBILITU S VAŠOU APLIKÁCIOU**

Niektoré webové aplikácie, REST API alebo pluginy využívajú metódy `PUT` alebo `DELETE`. Pred nasadením toto nastavenie otestujte. Ak vaša aplikácia vyžaduje ďalšie metódy, pridajte ich do zoznamu za `limit_except`.

6 Obmedzenie veľkosti požiadaviek

Predvolená maximálna veľkosť tela HTTP požiadavky je v NGINX 1 MB. Príliš vysoký limit umožňuje zahltiť server opakovanými veľkými požiadavkami, nastavte hodnotu podľa skutočnej potreby vašej aplikácie.

Otvorte súbor `/etc/nginx/nginx.conf`. Nájdite existujúci blok `http { ... }` a pridajte:

```
http {
    client_max_body_size 10m; # Upravte podľa potrieb aplikácie
    # ... zvyšok konfigurácie
}
```

Pre stránky bez nahrávania súborov odporúčame hodnotu `1m` alebo nižšiu. Pre WordPress alebo aplikácie s nahrávaním nastavte hodnotu podľa maximálnej očakávanej veľkosti súboru.

7 Nastavenie timeoutov

Správne nastavenie timeoutov chráni server pred útokmi, pri ktorých útočník zámerne spomaľuje posielanie požiadaviek, aby obsadil serverové spojenia.

V súbore `/etc/nginx/nginx.conf`, v bloku `http { ... }`, pridajte:

```
client_header_timeout 10s;
client_body_timeout   10s;
keepalive_timeout     65s;
send_timeout          10s;
```

Uvedené hodnoty sú bezpečným základom pre väčšinu stránok a zvyčajne nie je potrebné ich meniť. Ak vaša aplikácia spracováva nahrávanie veľkých súborov, zvýšte `client_body_timeout` podľa potreby.

02

KAPITOLA DRUHÁ

SSL/TLS a HTTPS

Šifrovanie komunikácie medzi serverom a návštevníkmi je dnes nevyhnutnosťou. Táto kapitola vás prevedie získaním certifikátu, správnou konfiguráciou TLS a automatickou obnovou certifikátu.

8 Získanie certifikátu cez Let's Encrypt

Let's Encrypt poskytuje bezplatné SSL/TLS certifikáty automaticky overené pre vašu doménu. Na ich získanie a správu použijeme nástroj Certbot.

Inštalácia Certbot

```
sudo apt install certbot python3-certbot-nginx -y
```

Získanie a inštalácia certifikátu

Nahraďte `vasa-stranka.sk` názvom vašej domény. Certbot automaticky upraví konfiguráciu v súbore `/etc/nginx/sites-available/vasa-stranka`:

```
sudo certbot --nginx -d vasa-stranka.sk -d www.vasa-stranka.sk
```

Certbot sa opýta na e-mailovú adresu pre upozornenia a požiada o súhlas s podmienkami používania. Po úspešnom dokončení je HTTPS aktívne a Certbot do vášho virtual hostu automaticky doplní SSL direktívy a vytvorí redirect blok pre port 80.

> 📋 **ČO CERTBOT PRIDÁ DO VAŠEJ KONFIGURÁCIE**
>
> Certbot do bloku `server { ... }` pre port 443 pridá direktívy `ssl_certificate`, `ssl_certificate_key` a riadok `include /etc/letsencrypt/options-ssl-nginx.conf`. Tento include súbor už obsahuje nastavenia protokolov TLS, cipher suites a session cache, nie je potrebné ich pridávať manuálne. Manuálne pridanie duplicitných direktív spôsobí chybu pri načítaní konfigurácie.

Overenie obsahu Certbot SSL konfigurácie

Pred akýmikoľvek manuálnymi úpravami SSL nastavení si overte, čo Certbot už nakonfiguroval:

```
cat /etc/letsencrypt/options-ssl-nginx.conf
```

Výstup ukáže, ktoré protokoly a cipher suites sú už nastavené. Certbot štandardne povolí TLS 1.2 a 1.3 a zakáže staršie verzie, čo je správne nastavenie bez potreby ďalších zmien.

> ⚠️ **NEPRIDÁVAJTE SSL DIREKTÍVY MANUÁLNE PO CERTBOTE**
>
> Ak Certbot váš virtual host už nakonfiguroval, nepridávajte manuálne direktívy `ssl_protocols`, `ssl_ciphers`, `ssl_prefer_server_ciphers` ani `ssl_session_*`. Sú už nastavené cez include súbor. Duplicitné direktívy spôsobia chybu `[emerg] directive is duplicate` a NGINX odmietne načítať konfiguráciu.

9 Povolenie HSTS

HTTP Strict Transport Security (HSTS) inštruuje prehliadač, aby sa na vašu stránku vždy pripájal výhradne cez HTTPS. Certbot automaticky vytvára redirect z portu 80 na 443, HSTS ide o krok ďalej a zabráni prehliadaču odoslať HTTP požiadavku vôbec, bez čakania na redirect zo servera.

Otvorte súbor `/etc/nginx/sites-available/vasa-stranka`. V bloku `server { ... }` pre port 443, za SSL direktívami a pred `location` blokmi, pridajte:

```
add_header Strict-Transport-Security "max-age=63072000; includeSubDomains; preload" always;
```

> ⚠️ **HSTS JE ŤAŽKO ZVRATITEĽNÉ**
>
> Po aktivácii HSTS prehliadače odmietnu načítať stránku cez HTTP počas celej doby platnosti definovanej v `max-age` (tu 2 roky). 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.

10 Automatická obnova certifikátu

Certbot pri inštalácii automaticky vytvorí systemd timer pre obnovu certifikátu. Certifikáty Let's Encrypt sú platné 90 dní. Certbot ich obnoví automaticky keď im zostávajú menej ako 30 dní platnosti.

Overte, že timer je aktívny:

```
sudo systemctl status certbot.timer
```

Výstup by mal obsahovať `Active: active (waiting)`. Funkčnosť obnovy otestujte bez skutočnej zmeny certifikátu:

```
sudo certbot renew --dry-run
```

Ak príkaz prebehne bez chýb, automatická obnova je správne nakonfigurovaná.

11 Overenie SSL konfigurácie

Po dokončení konfigurácie SSL/TLS overte jej kvalitu pomocou online nástroja SSL Labs:

1. Navštívte **ssllabs.com/ssltest**
2. Zadajte adresu vašej domény
3. Počkajte na dokončenie testu (1–2 minúty)
4. Cieľom je hodnotenie **A** alebo **A+**

> ✅ **BEŽNÉ PRÍČINY HODNOTENIA NIŽŠIEHO AKO A**
>
> Najčastejšie ide o povolenie starých protokolov (TLS 1.0/1.1), slabé cipher suites alebo chýbajúci HSTS. Správa SSL Labs presne identifikuje problém a jeho príčinu. Ak ste postupovali podľa tohto návodu a použili Certbot, hodnotenie A by malo byť dosiahnuteľné bez ďalších úprav.

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 nežiaduce vkladanie obsahu z cudzích zdrojov.

12 Základné bezpečnostné hlavičky

Otvorte súbor `/etc/nginx/sites-available/vasa-stranka`. Všetky hlavičky pridajte do bloku `server { ... }` pre port 443, za SSL direktívami a pred `location` blokmi. Štruktúra bloku by mala vyzerať takto:

```
server {
    listen 443 ssl;
    server_name vasa-stranka.sk;

    # SSL direktívy (pridané Certbotom)
    ssl_certificate ...;
    ssl_certificate_key ...;
    include /etc/letsencrypt/options-ssl-nginx.conf;

    # Bezpečnostné hlavičky patria sem
    add_header Strict-Transport-Security "max-age=63072000; includeSubDomains; preload" always;
    add_header X-Frame-Options "DENY" always;
    add_header X-Content-Type-Options "nosniff" always;
    add_header Referrer-Policy "strict-origin-when-cross-origin" always;
    add_header Permissions-Policy "geolocation=(), microphone=(), camera=(), payment=()" always;
    add_header Cross-Origin-Opener-Policy "same-origin" always;
    add_header Cross-Origin-Resource-Policy "same-origin" always;

    # Location bloky nasledujú po hlavičkách
    location / {
        ...
    }
}
```

Hlavička	Účel
`X-Frame-Options: DENY`	Zabraňuje načítaniu stránky v iframe. 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)
`Cross-Origin-*`	Izoluje stránku od cross-origin útokov

> ⚠️ **HLAVIČKU X-XSS-PROTECTION NEPOUŽÍVAJTE**
>
> Táto hlavička je zastaraná, moderné prehliadače ju nepodporujú a v niektorých prípadoch môže sama o sebe zraniteľnosti spôsobiť. Vynechajte ju úplne, alebo jej priraďte hodnotu `0`.

13 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. Nesprávne nastavená CSP hlavička môže narušiť funkčnosť stránky, preto budeme postupovať vo dvoch fázach.

Obe direktívy patria do bloku `server { ... }` pre port 443 v súbore `/etc/nginx/sites-available/vasa-stranka`, spolu s ostatnými hlavičkami.

Fáza 1 - Sledovanie bez blokovania

Začnite s hlavičkou `Content-Security-Policy-Report-Only`. Porušenia len zaznamenáva do konzoly prehliadača (F12 → Console) bez toho, aby čokoľvek zablokovala. Používajte ju 2–4 týždne a sledujte výstupy.

```
# Fáza 1 - len sledovanie, nič sa neblokuje
add_header Content-Security-Policy-Report-Only "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:;" always;
```

Fáza 2 - Nasadenie ostrej politiky

Po otestovaní nahraďte hlavičku ostrou verziou:

```
# Fáza 2 - po otestovaní nahraďte ostrou verziou
add_header 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:;" always;
```

⚠️ **'UNSAFE-INLINE' OSLABUJE OCHRANU PRED XSS**

Hodnoty `'unsafe-inline'` a `'unsafe-eval'` sú často nevyhnutné pre správne fungovanie CMS alebo pluginov, avšak znižujú ochranu pred XSS útokmi. Cieľom je ich postupné odstránenie. Hodnotu `'unsafe-eval'` skúste odstrániť ako prvú. Väčšina moderných stránok ju nevyžaduje.

14 Overenie bezpečnostných hlavičiek

Po nasadení hlavičiek overte ich správnosť. Najprv skontrolujte, že NGINX 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. Ak ste stránku testovali predtým, výsledok môže byť z cache, počkajte chvíľu a skúste znova

✅ **CIEĽOVÉ HODNOTENIE**

Na **securityheaders.com** je cieľom hodnotenie **A** alebo **A+**. Na **observatory.mozilla.org** cieľte aspoň na skóre **50 bodov**. Hodnotenie sa zlepší po nasadení CSP v ostrej verzii (Fáza 2).

04

KAPITOLA ŠTVRTÁ

Obmedzenie prístupu

Nastavenia, ktoré určujú, kto a čo má prístup k vášmu serveru.

15 Rate limiting. Ochrana pred brute-force útokmi

Rate limiting obmedzuje počet požiadaviek, ktoré môže jeden klient odoslať za určitý čas.

Krok 1 - Definícia zón v hlavnej konfigurácii

Otvorte súbor `/etc/nginx/nginx.conf`. Do bloku `http { ... }` pridajte definície zón:

```
# Zóna pre všeobecné požiadavky: 10 req/s na IP, pamäť 10MB
limit_req_zone $binary_remote_addr zone=general:10m rate=10r/s;

# Zóna pre prihlasovacie formuláre: 1 req/s na IP
limit_req_zone $binary_remote_addr zone=login:10m rate=1r/s;
```

Krok 2 - Aplikácia zón vo virtual hoste

Otvorte súbor `/etc/nginx/sites-available/vasa-stranka` a aplikujte zóny na príslušné `location` bloky. Tieto bloky musia byť definované **pred** blokom `location / { ... }`:

```
# Ochrana prihlasovacej stránky (WordPress)
location = /wp-login.php {
    limit_req zone=login burst=3 nodelay;
    limit_req_status 429;
}
# Všeobecný rate limiting pre celú stránku
location / {
    limit_req zone=general burst=20 nodelay;
    limit_req_status 429;
    # ... zvyšok konfigurácie
}
```

📋 **ČO ZNAMENAJÚ JEDNOTLIVÉ PARAMETRE**

`rate=1r/s` - maximálne 1 požiadavka za sekundu. `burst=3` povolí krátkodobé prekročenie limitu o 3 požiadavky (pre bežné správanie prehliadačov). `nodelay` požiadavky v rámci burst limitu sa spracujú okamžite. `limit_req_status 429` pri prekročení limitu vráti štandardný HTTP kód *Too Many Requests*.

✅ **LADENIE HODNÔT PODĽA VAŠEJ PREVÁDZKY**

Uvedené hodnoty sú konzervatívnym základom pre väčšinu menších stránok. Ak legitímni používatelia dostávajú chybu 429, zvýšte hodnotu `burst` pre príslušnú zónu. Hodnotu `rate` meňte opatrne. Príliš vysoká hodnota znižuje ochranu pred brute-force útokmi. Priebeh blokovania sledujte príkazom: `sudo tail -f /var/log/nginx/error.log`

16 Obmedzenie prístupu podľa IP adresy

Ak sa administrátori pripájajú z pevných IP adries alebo vnútornej siete, obmedzte prístup k citlivým cestám výhradne na tieto adresy.

V súbore `/etc/nginx/sites-available/vasa-stranka`, pred blokom `location / { ... }`, pridajte:

```
location /admin {
    allow 192.168.1.0/24;  # Vaša vnútorná sieť
    allow 203.0.113.10;    # Konkrétna IP adresa administrátora
    deny all;
}
```

17 Ochrana pred brute-force útokmi - fail2ban

Nástroj `fail2ban` sleduje systémové logy a automaticky blokuje IP adresy, ktoré vykazujú známky útoku, napríklad opakované neúspešné pokusy o prihlásenie cez SSH alebo webový server. Na rozdiel od rate limitingu (sekcia 15), ktorý spomaľuje útočníka, fail2ban ho úplne zablokuje.

Inštalácia

```
sudo apt install fail2ban -y
```

Základná konfigurácia

Nikdy neupravujte súbor `/etc/fail2ban/jail.conf` priamo, keďže pri aktualizácii fail2ban sa prepíše. Vytvorte vlastný konfiguračný súbor `/etc/fail2ban/jail.local`, ktorý má prednosť pred predvolenou konfiguráciou:

```
sudo nano /etc/fail2ban/jail.local
```

Vložte nasledujúci obsah, pokrýva ochranu SSH aj NGINX:

```
[DEFAULT]
# Čas blokovania IP adresy (10 minút)
bantime  = 10m
# Časové okno pre sledovanie pokusov
findtime = 10m
# Počet neúspešných pokusov pred zablokovaním
maxretry = 5
# Backend pre sledovanie logov
backend  = systemd

[sshd]
enabled = true
port    = ssh

[nginx-http-auth]
enabled  = true
port     = http,https
logpath  = /var/log/nginx/error.log

[nginx-limit-req]
enabled  = true
port     = http,https
logpath  = /var/log/nginx/error.log
maxretry = 10
```

Spustenie a povolenie služby

```
sudo systemctl enable --now fail2ban
```

✅ **OVERENIE FUNKČNOSTI**

Zobrazte stav blokovania:

```
sudo fail2ban-client status
sudo fail2ban-client status sshd
```

Výstup ukáže počet zablokovaných IP adries a celkový počet zachytených pokusov. Ak jail `sshd` funguje, fail2ban je aktívny.

📋 **ODBLOKOVANIE VLASTNEJ IP ADRESY**

Ak omylom zablokujete vlastnú IP adresu, odblokujte ju príkazom. Nahraďte `1.2.3.4` vašou IP adresou:

```
sudo fail2ban-client set sshd unbanip 1.2.3.4
```

18 Konfigurácia firewallu

Firewall určuje, ktoré porty a služby sú dostupné z internetu. Bez firewallu je každý port na vašom serveri potenciálne verejne dostupný, aj služby, ktoré ste neplánovali zverejniť. 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.

Debian / Ubuntu - ufw

```
sudo apt install ufw -y
```

Povoľte potrebné porty v tomto poradí:

```
# 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 po povolení portov 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.

> 📋 **AK POUŽÍVATE INÝ SSH PORT**
>
> Ak ste SSH presunuli na iný port (napríklad 2222), nahraďte `22/tcp` správnym číslom: `sudo ufw allow 2222/tcp`. Predvolené pravidlo `allow 22/tcp` v takom prípade vynechajte.

RHEL - firewalld

Na RHEL je `firewalld` predvolene nainštalovaný a aktívny. Overte stav a pridajte potrebné služby:

```
# Overenie stavu
sudo systemctl status firewalld

# Povolenie HTTP, HTTPS a SSH (SSH býva povolené predvolene)
sudo firewall-cmd --permanent --add-service=http
sudo firewall-cmd --permanent --add-service=https
sudo firewall-cmd --permanent --add-service=ssh

# Načítanie pravidiel
sudo firewall-cmd --reload

# Overenie aktívnych pravidiel
sudo firewall-cmd --list-all
```

19 Blokovanie citlivých súborov a ciest

Nasledujúce bloky patria do súboru `/etc/nginx/sites-available/vasa-stranka`, do bloku `server { ... }` pre port 443. Je kriticky dôležité, aby boli definované **pred** blokom `location / { ... }`. NGINX vyhodnocuje regex `location` bloky v poradí ich výskytu a blok `location /` by ich inak predbehol.

Blokovanie skrytých súborov a adresárov

```
# Blokuj .git, .env, .htaccess a všetky ostatné skryté súbory
location ~ /\. {
    deny all;
    access_log off;
    log_not_found off;
}
```

Blokovanie citlivých súborov podľa prípony

```
# Blokuj zálohy, konfigurácie, skripty a logy
location ~* \.(bak|conf|dist|env|fla|inc|ini|log|psd|sh|sql|swp)$ {
    deny all;
    access_log off;
    log_not_found off;
}
```

Blokovanie XML-RPC *(len pre WordPress inštalácie)*

Ak váš server neprevádzkuje WordPress, tento blok vynechajte.

```
location = /xmlrpc.php {
    deny all;
    access_log off;
    log_not_found off;
}
```

✅ **OVERENIE BLOKOVANIA**

Po reloade otestujte blokovanie citlivých súborov:

```
curl -I https://vasa-stranka.sk/.env
curl -I https://vasa-stranka.sk/config.bak
```

Odpoveď musí byť `403 Forbidden`. Ak dostanete `200 OK`, skontrolujte poradie `location` blokov. Bloky so skrytými a príponami musia byť pred `location /`.

20 Konfigurácia logovania

Na Ubuntu a Debian je základné logovanie NGINX aktívne predvolene. Overte, že funguje správne, a nastavte vhodnú úroveň logovania chýb.

Umiestnenie a overenie logov

NGINX predvolene zapisuje logy do:

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

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

Úroveň logovania chýb

V súbore `/etc/nginx/nginx.conf`, v bloku `http { ... }`, nastavte úroveň `warn`. Zaznamenáva varovania aj chyby bez zbytočného šumu:

```
error_log /var/log/nginx/error.log warn;
```

Rotácia a oprávnenia logov

Rotácia logov je na Ubuntu a Debian predvolene nakonfigurovaná. Overte, že konfigurácia existuje:

```
cat /etc/logrotate.d/nginx
```

Ak súbor existuje s direktívami `rotate`, `daily` a `compress`, nie je potrebné nič meniť. Oprávnenia log súborov overte príkazom:

```
ls -la /var/log/nginx/
```

Štandardné oprávnenia sú `640`, teda čitateľné len pre `root` a skupinu `adm`. Toto je správne nastavenie.

05

KAPITOLA PIATA

Priebežná údržba

Bezpečnosť nie je jednorazový úkon. Tieto návyky a postupy udržujú váš server v bezpečnom stave a umožňujú rýchlu obnovu v prípade incidentu.

21 Testovanie konfigurácie pred každým reloadom

Toto je najdôležitejší návyk pri správe NGINX. Príkaz `nginx -t` overí syntaktickú správnosť konfigurácie bez reštartu služby. Ak nájde chybu, server zostane funkčný so starým nastavením.

```
# Vždy najprv otestujte
sudo nginx -t

# Až po úspešnom teste načítajte konfiguráciu
sudo systemctl reload nginx
```

Úspešný výstup vyzerá takto:

```
nginx: the configuration file /etc/nginx/nginx.conf syntax is ok
nginx: configuration file /etc/nginx/nginx.conf test is successful
```

⚠ **RELOAD VS. RESTART**

Používajte `systemctl reload nginx`, nie `restart`. Reload načíta novú konfiguráciu bez prerušenia aktívnych spojení. Restart celú službu zastaví a znova spustí. Počas toho je server nedostupný.

22 Pravidelná aktualizácia NGINX

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

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

> ✅ **AUTOMATICKÉ AKTUALIZÁCIE**
>
> Pre automatickú aplikáciu bezpečnostných záplat zvážte nasadenie nástroja **unattended-upgrades**, ktorý je podrobne opísaný v samostatnom manuáli. Pre menšie organizácie bez dedikovaného IT personálu je to jeden z najúčinnejších krokov na zníženie rizika.

23 Zálohovanie konfigurácie

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

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

# Záloha konkrétneho virtual hostu pred úpravou
sudo cp /etc/nginx/sites-available/vasa-stranka \
        /etc/nginx/sites-available/vasa-stranka.bak.$(date +%Y%m%d)
```

Ak ste oboznámení s nástrojom Git, konfiguráciu NGINX v ňom môžete verzionovať. Každá zmena je zaznamenaná a je možné sa vrátiť k predchádzajúcemu stavu. Repozitár musí byť privátny.

24 Plán pravidelného auditu

Pravidelný audit zabezpečuje, že bezpečnosť zostáva na požadovanej úrovni aj ako sa server a tím menia v čase.

Týždenné úlohy

- Kontrola chybového logu: `sudo tail -100 /var/log/nginx/error.log`
- Prehľad prístupových logov. Podozrivo vysoký počet požiadaviek z jednej IP, skenovanie citlivých ciest
- Overenie platnosti SSL certifikátu: `sudo certbot certificates`

Mesačné úlohy

- Aktualizácia NGINX a systémových balíkov
- Overenie bezpečnostných hlavičiek: **securityheaders.com**
- Overenie SSL konfigurácie: **ssllabs.com/ssltest**
- Overenie funkčnosti automatickej obnovy certifikátu: `sudo certbot renew --dry-run`

- Kontrola oprávnení log súborov: `ls -la /var/log/nginx/`

Ročné úlohy

- Komplexný audit konfigurácie. Porovnajte aktuálny stav s odporúčaniami tohto manuálu
- Revízia pravidiel rate limitingu a IP obmedzení podľa aktuálnych potrieb
- Penetračný test, ak to odôvodňuje profil rizík organizácie
- Kontrola súladu s legislatívou a bezpečnostnou politikou organizácie

> ✅ **DOKUMENTUJTE ZMENY**
>
> Každú zmenu konfigurácie zaznamenajte. Čo ste zmenili, kedy a prečo. Stačí jednoduchý textový súbor alebo tabuľka. Pri riešení incidentu alebo odovzdávaní správy servera inej osobe je táto dokumentácia neoceniteľná.

Čo ďalej

Dokončením tohto manuálu máte funkčný NGINX server s 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* (Debian/Ubuntu) alebo *dnf-automatic* (RHEL), ktoré sú podrobne opísané v samostatnom manuáli.
- **WordPress na NGINX** - ak prevádzkujete WordPress, pozrite si náš WordPress Hardening Manuál. Kroky v ňom sú navrhnuté ako doplnok k tomuto manuálu a spoločne tvoria komplexnú ochranu.
- **Pravidelné skenovanie** - VJ CSIRT poskytuje službu skenovania zraniteľností pre registrované organizácie na adrese **csirt.sk**.

A

PRÍLOHA

Vzorové konfigurácie

/etc/nginx/nginx.conf

```
user www-data;
worker_processes auto;
pid /run/nginx.pid;
include /etc/nginx/modules-enabled/*.conf;

events {
    worker_connections 768;
}

http {
    # Základné nastavenia
    sendfile on;
    tcp_nopush on;
    types_hash_max_size 2048;
    server_tokens off;

    include /etc/nginx/mime.types;
    default_type application/octet-stream;

    # Veľkosť požiadaviek a timeouty
    client_max_body_size 10m;
    client_header_timeout 10s;
    client_body_timeout   10s;
    keepalive_timeout     65s;
    send_timeout          10s;
    # Rate limiting zóny
    limit_req_zone $binary_remote_addr zone=general:10m rate=10r/s;
    limit_req_zone $binary_remote_addr zone=login:10m rate=1r/s;
```

```
    # Logovanie
    access_log /var/log/nginx/access.log;
    error_log  /var/log/nginx/error.log warn;

    # Gzip komprimácia
    gzip on;

    # Načítanie virtual hostov
    include /etc/nginx/conf.d/*.conf;
    include /etc/nginx/sites-enabled/*;
}
```

/etc/nginx/sites-available/vasa-stranka

```
server {
    server_name vasa-stranka.sk www.vasa-stranka.sk;
    root /var/www/vasa-stranka;
    index index.html index.htm;

    # SSL certifikáty doplnené Certbotom
    listen [::]:443 ssl ipv6only=on;
    listen 443 ssl;
    ssl_certificate /etc/letsencrypt/live/vasa-stranka.sk/fullchain.pem;
    ssl_certificate_key /etc/letsencrypt/live/vasa-stranka.sk/privkey.pem;
    include /etc/letsencrypt/options-ssl-nginx.conf;
    ssl_dhparam /etc/letsencrypt/ssl-dhparams.pem;

    # Bezpečnostné hlavičky
    add_header Strict-Transport-Security "max-age=63072000; includeSubDomains; preload" always;
    add_header X-Frame-Options "DENY" always;
    add_header X-Content-Type-Options "nosniff" always;
    add_header Referrer-Policy "strict-origin-when-cross-origin" always;
    add_header Permissions-Policy "geolocation=(), microphone=(), camera=(), payment=()" always;
    add_header Cross-Origin-Opener-Policy "same-origin" always;
    add_header Cross-Origin-Resource-Policy "same-origin" always;
    add_header 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:;" always;
```

```
    # Vlastná chybová stránka
    error_page 404 /404.html;
    location = /404.html { internal; }

    # Blokovanie citlivých súborov. MUSÍ byť pred location /
    location ~ /\. {
        deny all;
        access_log off;
        log_not_found off;
    }

    location ~* \.(bak|conf|dist|env|fla|inc|ini|log|psd|sh|sql|swp)$ {
        deny all;
        access_log off;
        log_not_found off;
    }

    # WordPress XML-RPC. Odstráňte ak WordPress nepoužívate
    location = /xmlrpc.php {
        deny all;
        access_log off;
        log_not_found off;
    }

    # Hlavný location blok
    location / {
        autoindex off;
        try_files $uri $uri/ =404;
        limit_req zone=general burst=20 nodelay;
        limit_req_status 429;
        limit_except GET POST HEAD {
            deny all;
        }
    }
}

# HTTP → HTTPS redirect, doplnené Certbotom
server {
    listen 80;
    listen [::]:80;
    server_name vasa-stranka.sk www.vasa-stranka.sk;
```

```
    if ($host = www.vasa-stranka.sk) {
        return 301 https://$host$request_uri;
    }
    if ($host = vasa-stranka.sk) {
        return 301 https://$host$request_uri;
    }
    return 404;
}
```

⚠️ **VZOROVÉ SÚBORY SÚ VÝCHODISKOVÝM BODOM, NIE HOTOVOU KONFIGURÁCIOU**

Nahraďte `vasa-stranka.sk` svojou doménou, upravte cesty k certifikátom a `root` adresár podľa vašej inštalácie. Hodnotu `client_max_body_size` prispôsobte potrebám vašej aplikácie. CSP politiku upravte po absolvovaní Fázy 1 (sekcia 13). Vždy spustite `sudo nginx -t` pred každým reloadom.