LINUX RENDSZERFEJLESZTÉS ÉS ÜZEMELTETÉS

5. forduló

ONE IDENTITY by Quest

A kategória támogatója: One Identity - Quest Hungary

Ismertető a feladathoz

Felhasznált idő: 00:00/20:00 **Elért pontszám:** 0/15

1. feladat 0/5 pont

Ubuntu Bionic Beaver -> Focal Fossa upgrade után nem indul el az nginx! Ez azért furcsa, mert a konfigurációd majdhogynem teljesen megegyezik a disztribúció által szállított alapbeállításokkal, a lényeges rész csupán ennyi:

```
server {
        listen 443 ssl default_server;
        ssl_certificate /etc/ssl/certs/server.crt;
        ssl_certificate_key /etc/ssl/private/server.key;
        root /var/www/html;
        index index.html
        server_name _;
        location / {
                try_files $uri $uri/ =404;
        }
}
```

Ugyanez a konfiguráció az upgrade előtt hiba nélkül működött. Mégsem indul el, a 443-as porton nem figyel semmi. **Mi lehet az oka?**

Válasz

- ○ Az Ubuntu Focal Fossa-ban hibás a default nginx csomag, és nem indul el, ha az SSL modul be van kapcsolva. A megoldást a modul kikapcsolása jelenti, vagy upgrade-elni kell az nginx-et az "NGINX Stable" PPA repository-ban levő verzióra.
- ● Az Ubuntu Focal Fossa-val szállított openssl-ben a SHA1-es aláírást használó tanúsítványok nem engedélyezettek az alapértelmezett biztonsági szinten (@SECLEVEL=2). Az nginx indulás előtt ellenőrzi az SSL tanúsítványai megfelelőségét, és nem fogadja el a szervertanúsítványt.
  Ez a válasz helyes, de nem jelölted meg.
- ○ Az Ubuntu Focal Fossa-val szállított nginx megköveteli, hogy az SSL privát kulcsot csak a root felhasználó olvashassa. Az lehet a gond, hogy a */etc/ssl/private/server.key* a root csoport által is olvasható.
- ○ A tanúsítványban levő publikus kulcs nem a megadott privát kulcshoz tartozik.
- ○ Az újabb nginx változatoknál kötelező explicit megadni a server_name paramétert, ha SSL-t akarunk használni.

Magyarázat

Egyfelől érdemes tudni, hogy a Debian Buster, ill. az Ubuntu Focal kiadástól kezdve az OpenSSL alapértelmezett biztonsági szintje (SECLEVEL) 1-ről 2-re változott. (https://www.debian.org/releases/buster/amd64/release-notes/ch-information.en.html#openssl-defaults) Ez a beállítás befolyásolja azt, hogy az OpenSSL-t használó alkalmazások milyen kulcsméretet és algoritmusokat fogadnak el biztonságosnak, bővebb információkat lásd pl. itt: https://manpages.debian.org/bullseye/libssl-doc/SSL_CTX_set_security_level.3ssl.en.html.

Másfelől pedig az NGINX úgy működik, hogy a beolvasott tanúsítványokat megpróbálja betölteni egy SSL context-be (https://github.com/nginx/nginx/blob/master/src/event/ngx_event_openssl.c#L441), és ha ez nem sikerül, akkor egy ehhez hasonló hibaüzenetet kapunk:

```
~$ systemctl status nginx.service

● nginx.service - A high performance web server and a reverse proxy server

     Loaded: loaded (/lib/systemd/system/nginx.service; enabled; vendor preset: enabled)

     Active: failed (Result: exit-code) since Wed 2022-08-17 09:22:16 CEST; 21s ago

       Docs: man:nginx(8)

    Process: 1709 ExecStartPre=/usr/sbin/nginx -t -q -g daemon on; master_process on; (code=exited, status=1/



Aug 17 09:22:16 test-ssh systemd[1]: nginx.service: Succeeded.

Aug 17 09:22:16 test-ssh systemd[1]: Stopped A high performance web server and a reverse proxy server.

Aug 17 09:22:16 test-ssh systemd[1]: Starting A high performance web server and a reverse proxy server...

Aug 17 09:22:16 test-ssh nginx[1709]: nginx: [emerg] SSL_CTX_use_certificate("/etc/ssl/certs/server.crt") fai

Aug 17 09:22:16 test-ssh nginx[1709]: nginx: configuration file /etc/nginx/nginx.conf test failed

Aug 17 09:22:16 test-ssh systemd[1]: nginx.service: Control process exited, code=exited, status=1/FAILURE

Aug 17 09:22:16 test-ssh systemd[1]: nginx.service: Failed with result 'exit-code'.

Aug 17 09:22:16 test-ssh systemd[1]: Failed to start A high performance web server and a reverse proxy server
```

Ugyanilyen hibát (csak másik OpenSSL hibaüzenetet) kapnánk akkor is, ha pl. 1024 bites RSA kulcsot használnánk a példában.

Érdemes még tudni, hogy OpenSSL 3-tól (pl. Ubuntu Jammy Jellyfish vagy későbbi) már 1-es biztonsági szinten sem használható az SHA1, valamint a TLS 1.1 és alatti protokollok
sem! (https://www.openssl.org/docs/manmaster/man3/SSL_CTX_set_security_level.html)

"A tanúsítványban levő publikus kulcs nem a megadott privát kulcshoz tartozik." megoldás azért nem jó, mert a feladat kiírásban szerepelt, hogy ugyanez a konfiguráció Bionic Beaver-en működött.

2. feladat 0/0 pont

Az 1024-bites RSA/DSA kulcsok, illetve az SHA1-es aláírást használó tanúsítványok általában nem tekinthetőek biztonságosnak 2022-ben.

Milyen esetben **NEM MUSZÁJ** lecserélnünk a webszerver által használt tanúsítványt és/vagy tanúsítványláncot, amennyiben azt szeretnénk, hogy a weboldalunk biztonsági figyelmeztetés nélkül töltődjön be az elterjedt webböngészőkkel? Feltételezzük, hogy a self-signed Root CA közvetlenül írja alá a szerverünk tanúsítványát!

Válasz

- ( ) Ha a Root CA tanúsítványa erős (minimum 4096-bit RSA, SHA-512 hash-t használó aláírással), akkor a szervertanúsítvány tartalmazhat SHA1-es hash-t használó aláírást
- ( ) Ha a szervertanúsítvány erős (minimum 2048-bit RSA, SHA-256 hash-t használó aláírással), akkor a Root CA kulcsa lehet 1024 bites RSA is, feltéve, hogy a Root CA tanúsítvány aláírása SHA-256 vagy erősebb hash-t használ.
- (•) Ha a szervertanúsítvány erős (minimum 2048-bit RSA, SHA-256 hash-t használó aláírással), akkor a Root CA tanúsítványának aláírása használhat SHA1 hash-t, feltéve, hogy a kulcs 2048-bit RSA vagy erősebb.
  Ez a válasz helyes, de nem jelölted meg.
- ( ) Egy megbízható tanúsítványláncban sehol sem lehet 2048-bit RSA-nál kevésbé biztonságos kulcs, illetve sehol sem lehet benne SHA-256-nál kevésbé biztonságos hash, legalábbis a Firefox és a Chrome 2022. július 1-et megelőző legfrissebb verziói alapbeállítások mellett egyetlen támogatott operációs rendszer alatt sem fogadják el ezeket.

Magyarázat

Kedves Versenyzők!

A feladatot 0 pontosra állítottuk, ugyanis az "Egy megbízható tanúsítványláncban sehol sem lehet 2048-bit RSA-nál kevésbé biztonságos kulcs, illetve sehol sem lehet benne SHA-256-nál kevésbé biztonságos hash, legalábbis a Firefox és a Chrome 2022. július 1-et megelőző legfrissebb verziói alapbeállítások mellett egyetlen támogatott operációs rendszer alatt sem fogadják el ezeket." válaszlehetőség tévesen lett helyesnek jelölve. Így a feladat félrevezető módon check-box-ként jelent meg, holott csak egy helyes válasz van, tehát radio button-ként kellett volna megjeleníteni.

Elnézést kérünk a kellemetlenségért!

A szervezők

Az 1024 bites (aszimmetrikus) RSA kulcsméret körülbelül 80 bit szimmetrikus kulcsméret biztonságának felel meg. Ugyan véletlenszerű próbálkozással ilyen kulcsokat sem lehet belátható mennyiségű erőforrás használata mellett kitalálni, mégis egyetértés van az iparágban arról, hogy ez a kulcsméret már nem elegendő ahhoz, hogy ellenálljon pl. kormányok által finanszírozott, illetve bizonyos side-channel támadásoknak (pl. https://threatpost.com/libgcrypt-sliding-right-attack-allows-recovery-of-rsa-1024-keys/126675/).

Minden hash algoritmus úgy működik, hogy egy változó hosszú bemenetet egy fix hosszúságú kimenetre transzformál, így szükségképpen előfordul olyan, hogy két különböző bemenetnek ugyanaz a kimenete. A kriptográfiai hash-ek (mint pl. SHA1) azt hivatottak biztosítani, hogy ne lehessen egyszerűen előállítani olyan bemenetet, ami egy ismert kimenetet ad. Kb 2019-től ismert, hogy a SHA1 algoritmushoz generálható hash ütközés néhány tízezer dollár értékű számítási erőforrással.

A fentiek miatt 1024 bites RSA kulcsokat sem CA-k, sem szervertanúsítványok esetében nem biztonságos használni, ezért a 2. megoldás nem jó. Mivel SHA1 ütközés könnyen generálható, így egy ilyen tanúsítvány hamisítható is, ezért az 1. megoldás sem jó.

Ellenben a Root CA tanúsítványok esetén a megbízhatóság mindenképpen egy explicit döntés, mivel ezeket a tanúsítványokat a Root CA a saját privát kulcsával írja alá. Egy ilyen aláírást nincs értelme hamisítani, mert a hitelesség ellenőrzésében az aláírás nem játszik szerepet. A Root CA-k érvényességi ideje azonban jellemzően nagyon hosszú (tíz évnél több), és több olyan CA tanúsítvány van, amelyet még akkor állítottak ki, amikor az SHA1 algoritmus még biztonságosnak számított, ezért nincs értelme annak, hogy ezeket a CA-kban ne bízzanak meg a böngészők. Ezért jó az alábbi két megoldás:

- Ha a szervertanúsítvány erős (minimum 2048-bit RSA, SHA-256 hash-t használó aláírással), akkor a Root CA tanúsítványának aláírása használhat SHA1 hash-t, feltéve, hogy a kulcs 2048-bit RSA vagy erősebb.
- Egy megbízható tanúsítványláncban sehol sem lehet 2048-bit RSA-nál kevésbé biztonságos kulcs, illetve sehol sem lehet benne SHA-256-nál kevésbé biztonságos hash, legalábbis a Firefox és a Chrome 2022. július 1-et megelőző legfrissebb verziói alapbeállítások mellett egyetlen támogatott operációs rendszer alatt sem fogadják el ezeket.

3. feladat 0/10 pont

A lenti parancsot egy olyan környezetben szeretnéd működésre bírni, ahol:

• a "https://internal.server" egy elérhető intranet webszerver

• a */etc/ssl/certs* könyvtárba letöltötted a belső céges Root CA tanúsítványát PEM formátumban

• a Root CA írta alá az intranet webszerver tanúsítványát, a tanúsítványok érvényesek

• a disztribúciód a */etc/ssl/certs* alatt tartja a megbízható tanúítványkiadókat

• a disztribúciód által kiadott curl programot használod, amelyet OpenSSL-lel fordítottak

Az alábbi hibaüzenetet kapod:

```
$ curl https://internal.server/status
curl: (60) SSL certificate problem: unable to get local issuer certificate
```

Mi lehet a hiba?

Válaszok

- [ ] Nem lehet feloldani az internal.server névhez tartozó IP címet.
- [ ] A webszerver tanúsítványa nem az *internal.server* névhez van kiállítva.
- [x] Nem elég a Root CA tanúsítványát letölteni, hanem szükséges az is, hogy az OpenSSL a subject hash alapján megtalálja ezt a file-t a */etc/ssl/certs* könyvtárban. A szimbolikus link a *c_rehash* programmal is elkészíthető.
  Ez a válasz helyes, de nem jelölted meg.
- [ ] A Root CA kulcsa gyenge.
- [x] A szervertanúsítványt a Root CA-hoz tartozó másik kulccsal írták alá, így az szervertanúsítványban levő AuthorityKeyIdentifier és a Root CA SubjectKeyIdentifier kiterjesztése nem egyezik.
  Ez a válasz helyes, de nem jelölted meg.
- [ ] Nem elérhető a Root CA-hoz tartozó CRL, noha a szervertanúsítvány *cRLDistributionPoints* kiterjesztése nem üres.

Magyarázat

- "Nem lehet feloldani az internal.server névhez tartozó IP címet."
- "A webszerver tanúsítványa nem az *internal.server* névhez van kiállítva."
- "A Root CA kulcsa gyenge."

megoldások esetében másfajta hibaüzenetet kapnánk. A "Nem elérhető a Root CA-hoz tartozó CRL, noha a szervertanúsítvány *cRLDistributionPoints* kiterjesztése nem üres." megoldás azért nem jó, mert a *curl* nem tölt le visszavonási listát (CRL), legfeljebb egy már letöltött fájlt lehetne neki megadni a *--crlfile* opcióval, de ennek használata sem alapértelmezett.

A megbízható tanúsítványkiadók alapértelmezetten a ***/etc/ssl/certs*** *vannak. Azonban amikor egy tanúsítványról el kell dönteni, hogy megbízható-e, az openssl nem olvassa be az összes fájlt egyesével, mert az túlságosan erőforráspazarló lenne, hanem a Subject DN alapján egy hash-t készít, és az ebből képzett fájlnevet próbálja megnyitni. A c_rehash program szimbolikus linkeket hoz létre a tanúsítványokban levő subject alapján. (Ráadásul általában kettőt minden fájlra, mert létezik egy régi és egy új hash-elési eljárás is. Ha ugyanazt a subject-et több fájlban is megtalálja, akkor a .0 suffix inkrementálódik. Visszavonási listák esetén .r0 ... .rn suffix használatos; ezeket is a kiállító CA DN-je alapján keresi az openssl.)*

```
/etc/ssl/certs$ ls -l |grep ca.crt
lrwxrwxrwx 1 root root     6 Aug 16 14:47 c68853ad.0 -> ca.crt
-rw-r--r-- 1 root root  1513 Aug 16 14:46 ca.crt
lrwxrwxrwx 1 root root     6 Aug 16 14:47 efabc9f6.0 -> ca.crt
```

"A szervertanúsítványt a Root CA-hoz tartozó másik kulccsal írták alá, így az szervertanúsítványban levő AuthorityKeyIdentifier és a Root CA SubjectKeyIdentifier kiterjesztése nem egyezik." megoldás *is helyes, lásd man verify:*

The relevant authority key identifier components of the current certificate (if present) must match the subject key identifier (if present) and issuer and serial number of the candidate issuer

Megjelenés

Világos