LINUX RENDSZERFEJLESZTÉS ÉS ÜZEMELTETÉS

1. forduló

ONE IDENTITY by Quest

A kategória támogatója: One Identity - Quest Hungary

Ismertető a feladathoz

Kérjük, hogy a feladatlap indítása előtt mindenképp olvasd el az alábbi útmutatót:

- MINDEN kérdésre **van helyes válasz**.
- Olyan kérdés **NINCS**, amire az összes válasz helyes, ha mégis az összes választ bejelölöd, arra a feladatra automatikusan 0 pont jár.
- A **radio button-os kérdésekre egy helyes válasz van**.
- **Ha lejár a feladatlap ideje, a rendszer AUTOMATIKUSAN** beküldi azt az addig megjelölt válaszokkal.
- Azokat a feladatlapokat, amelyekhez **csatolmány** tartozik, javasoljuk **NEM mobilon** elindítani, erre az érintett feladatlapok előtt külön felhívjuk a figyelmet.
- Az **adatbekérős feladatokra NEM jár részpontszám**, csak a feleletválasztósakra.
- **Helyezéseket a 4. forduló után mutatunk**, százalékos formában: adott kategóriában a TOP 20-40-60%-hoz tartozol.
- **Badge-ket** szintén a 4.forduló után kapsz majd először.
- Ha egyszerre több böngészőből, több ablakban vagy több eszközről megnyitod ugyanazt a feladatlapot, **nem tudjuk vállalni** az adatmentéssel kapcsolatban esetlegesen felmerülő anomáliákért a felelősséget!
- A hét forduló során az egyes kategóriákban (de nem feltétlenül mindegyikben) **könnyű-közepes-nehéz kérdésekkel** egyaránt találkozhatsz majd.

Jó versenyzést kívánunk!

Felhasznált idő: 00:00/20:00 **Elért pontszám:** 0/12

Indítás utáni csatolmányok

1. feladat 0/2 pont

Van egy nagy SSD-m, amit még nagyobbra cserélnék, így eladásra kerül. Szeretnék biztos lenni abban, hogy nem marad rajta semmi olvasható, mivel sok fontos, magánjellegű file van rajta. Azt a megoldást találtam ki, hogy véletlenszerű byte-okkal írom tele többször, végtelen ciklusban. Este elindítom a scriptet, reggelre a sok felülírás miatt teljesen eltűnik minden adat róla.

Reggel leállítom a scriptet, és a biztonság kedvéért próbát teszek egy adatvisszaállító programmal, ami nagy meglepetésemre könnyedén megtalálja az elvileg letörölt adatokat, sérülés nélkül visszaállítva a file-okat.

Mi lehetett a baj?

A script:

```
while true ; do dd if=/dev/random of=/dev/sdb ; done
```

Válasz

- ○ Az SSD túl nagy, a dd a címzési megoldás miatt csak az első 32GB-t tudja írni, a fennmaradó részen a file-ok érintetlenek maradnak
- ○ Mivel létező filerendszer van az SSD-n, a dd visszautasítja az arra való írást
- ○ A /dev/random csak bash scriptek számára elérhető, a dd nem tudja olvasni
- ● A /dev/random nagyon lassú, így hiába futott egész éjjel a script, csak az SSD elejét tudta felülírni.
  Ez a válasz helyes, de nem jelölted meg.

Magyarázat

Sem a méret, sem a meglévő filerendszer nem probléma a dd-nek, és a /dev/random is elérhető bashből, csak éppen nagyon lassú. Helyette a /dev/urandom használata ajánlott, amivel remekül meg is oldható a feladat.

Alapvetően mindez az entropy hiány miatt van, amit az urandom úgy kezel, hogy kevésbé biztonságos. Létezik entropy generátor, ami ezt gyorsítani tudja.

2. feladat 0/5 pont

Szeretném az otthon folyamatosan bekapcsolt, Linuxot futtató Raspberry Pi serveremen 80-as porton futó weboldalt bármikor, bárhonnan elérni. Az internet szolgáltatóm 192.168.x.x formátumú intranetes IP címeket oszt ki a routernek, így kívülről, az internetről nem látható.

Van egy VPS-em is, amin szintén Linux fut, ez gond nélkül elérhető bárhonnan az "envpsem.com" címen.

Hogyan tudnám bármikor, bárhonnan megbízhatóan elérni a Raspberryn futó webservert az envpsem.com:8080 címen?

Hogy látható legyen a 8080 port, a VPS-en az sshd_configban a GatewayPorts opciót yes-re állítottuk.

Válasz

- ○ A Raspberryt beregisztráljuk egy dinamikus DNS szolgáltatóhoz (pl. noip.com), a rajta futó kliens segítségével a Raspberry mindig elérhető az általunk választott (pl. enraspberrym.ddns.com) címen. A VPS-ről felépítünk egy ssh tunnelt erre a címre:ssh -L *:8080:enraspberrym.ddns.com:80 -N enraspberrym.ddns.com.
- ○ A Raspberryről egy ssh tunnelt felépítve a VPS felé annak egy portját átirányítjuk a Raspberry 80-as portjára:ssh -L localhost:80:*:8080 -N envpsem.com.
- ○ A Raspberryről egy reverse ssh tunnelt felépítve a VPS felé annak egy portját átirányítjuk a Raspberry 22-es (ssh) portjára:ssh -R *:2222:localhost:22 envpsem.com Ezután az envpsem.com-ra ssh-n belépve felépítjük a tunnelt visszafelé:

  ssh -L *:8080:localhost:80 -N localhost -p 2222

  Az első, reverse tunnelt bontjuk, már nincs rá szükség a továbbiakban.
- ● A Raspberryről egy reverse ssh tunnelt felépítve a VPS felé annak egy portját átirányítjuk a Raspberry 80-as portjára:

  ssh -R *:8080:localhost:80 -N envpsem.com
  Ez a válasz helyes, de nem jelölted meg.

Magyarázat

Mivel kívülről nem látható az otthoni gép, mindenképpen reverse tunnelt kell felépíteni onnan kiindulva a VPS felé.

3. feladat 0/5 pont

Mert rendszerfejlesztőként fontos a ciklikus gondolkodás...Ebben a feladatban ez kerül előtérbe!

Rendszermérnökként a hálózat forgalmának és sebességének optimalizálása, a performancia javítása a feladatod, de néha a tűzfalszabályokat is kell módosítani, így elkerülhetetlen a hálózati forgalom elemzése is. Néhány szabály FIXME-zve van (#FIXME: tryharder), valószínű javítani kell rajtuk, mert nem elég szigorúak. Ahh, még egy feladat...

Egyik alkalommal, ahogy a hálózati forgalmakat nézed, találsz közöttük egy érdekes kimenő packet-et, amiben volt egy zip file. Kíváncsi leszel, mivel tudod, hogy a céges policy szabályozza a külső IP-re történő céges adatok továbbítását, illetve a céges adatok megosztására is van külön share. Ahogy jobban megnézed a forgalmakat, egyre furcsábbnak találod, hogy időnként újra előkerül ez a különös packet a zip file-al, viszont úgy döntesz, még nem jelented a manageredneknek, inkább beszerzel evidenciákat és mindent megteszel, hogy elcsípd. Nem sokkal később sikerül is elkapnod.

Vajon kitudod deríteni mi van a legmélyén?

Válasz

- ( ) .json file
- ( ) .sql file céges adatokkal
- ( ) .py file amit elindítva remote hozzáférést ad a céges hálózathoz
- ( ) .rules file, ami backdoor-t nyit a céges hálózaton
- (•) kiterjesztés nélküli állomány

Ez a válasz helyes, de nem jelölted meg.

Magyarázat

Ennek a tömörített állománynak a mélyén lesz egy file, aminek a neve – (kötőjel, dash). Ez valid filename, bár sok félreértésre ad okot Unix/Linux típusú rendszereken. Utóbbiaknál a – mint argumentum használata a STDIN/STDOUT -ra vonatkozik. Ennek a file-nak a megnyitásához konzolon a teljes PATH-t kell használni.

belépve a nested 5004.zip mappába:

```
while [ "`find . -type f -name '*.zip' | wc -l`" -gt 0 ]; do find -type f -name "*.zip" -exec unzip -- '{}' \; -exec rm -- '{}' \;; done
```