Linux rendszerfejlesztés és üzemeltetés (One Identity)
1. forduló

Ismertető a feladathoz

1.forduló

Tekintettel arra, hogy egy választ sem rögzítettél az alábbi feladatlapon, ebben a fordulóban a kitöltésére rendelkezésre álló idő teljes egésze, azaz 10 perc került rögzítésre mint megoldáshoz felhasznált idő.

1. feladat 0 / 5 pont

Elfelejtett root jelszó esetén hogyan lehet új root jelszót beállítani?

- [ ] A telepítéskor megadott biztonsági kód segítségével a root jelszó megváltoztatható
- [x] Megfelelő sudo jogok megléte esetén egy másik felhasználóval kicserélhető a root jelszó
- [ ] Ez csak a rendszer újratelepítésével orvosolható
- [ ] Az /etc/passwd fájl törlése után a rendszer új jelszót fog kérni induláskor
- [ ] A rendszer indításakor le kell nyomni az F7 gombot, hogy új jelszót állítsunk be
- [x] A consolról bármikor, csak 2 reboot kell, és kb.5 perc leállás

Magyarázat a megoldáshoz

A consolról bármikor, csak 2 reboot kell, és kb.5 perc leállás: A helyes lépéssor:
1, újraindítom a gépet.
2. a boot folyamatban a boot managerben módosítom a kernel paraméterek, hogy az init=/bin/bash
3. amikor megjön a prompt -itt ilyenkor nem kell felhasználónevet vagy jeszót beírni - , editálom a /etc/passwd vagy /etc/shadow file-okat, felülírva a root jelszót egy ismert jelszó hash-évél (attól függ hogy melyik, fileról van szó, hogy hogyan van konfigurálva a rendszer, nagy valószínűséggel az /etc/shadow file lesz a nyerő)
4. újraindítom a gépet megint és a loginnál már be lehet lépni az új jelszóval aminek a hash-ét az előbb elhejeztük az /etc/passwd vagy /etc/shadow fileban.

A 2. és a 3 lépés végrehajtása kb 5 perc, ha valaki nagyon lassan gépel, akkor max 10 perc.

2. feladat 0 / 5 pont

Mit csinál az alábbi bash script?

```
#!/bin/bash
for f in *
do
  if [[ -f $f ]]
  then
    echo $f: $(cat $f | wc -l)
  fi
done
```

- [ ] A többi válasz közül egyik sem helyes
- [ ] A szkript saját mappájából minden mappára kiírja, hogy hány fájl van benne
- [ ] A szkript saját mappájából minden fájlra kiírja, hogy hány sora van
- [ ] Az aktuális mappából minden mappára kiírja, hogy hány fájl van benne
- [ ] Az aktuális mappából minden fájlra kiírja, hogy hány sora van
- [x] Az aktuális mappából minden fájlra kiírja, hogy hány sora van (ha a fájlok nevében nincs pár speciális karakter)

Magyarázat a megoldáshoz

3. feladat 0 / 5 pont

A csapatodból többek szerint azért lassú az egyik szerver, mert rosszul lett particionálva. Stackoverflown azt olvasták, hogy particionálás során érdemes a partíciókat egész megabyte határokon kezdeni. Utánajártunk és kiderült, hogy pontosan a cylinderhatárról van szó.

A mellékelt információk alapján, megfelelően lett-e particionálva ez a lemez?

```
root@home:/dev# fdisk /dev/sda

Welcome to fdisk (util-linux 2.31.1).
Changes will remain in memory only, until you decide to write them.
Be careful before using the write command.


Command (m for help): p
Disk /dev/sda: 111.8 GiB, 120034123776 bytes, 234441648 sectors
Units: sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
Disklabel type: dos
Disk identifier: 0x446de33b

Device     Boot     Start       End   Sectors  Size Id Type
/dev/sda1          2048 213960703 213958656  102G 83 Linux
/dev/sda2       213960704 234440703  20480000  9.8G 82 Linux swap / Solaris

Command (m for help):
```

```
root@home:/home/endre# sudo lshw -class disk
  *-disk
       description: ATA Disk
       product: Samsung SSD 840
       physical id: 0.0.0
       bus info: scsi@0:0.0.0
       logical name: /dev/sda
       version: 8B6Q
       serial: S1D5NEBF230415P
       size: 111GiB (120GB)
       capabilities: partitioned partitioned:dos
       configuration: ansiversion=5 logicalsectorsize=512 sectorsize=512 signature=446de33b
```

- [x] Igen
- [ ] Valószínűleg, de a megadott információk alapján nem lehet biztosan eldönteni
- [ ] Nem, mert DOS módban lett particionálva
- [ ] Lényegtelen, mivel SSD esetében ez a szempont már nem releváns
- [x] A disk geomertiát is tudnom kellene a válaszhoz, de mivel ez a diszk úgy tűnik, mintha egy SSD lenne, a tapasztalatom azt súgja, hogy igen
- [ ] A /dev/sda1 partíció megfelel, de a /dev/sda2 nem
- [ ] A /dev/sda2 partíció megfelel, de a /dev/sda1 nem
- [x] A Stockoverflown lévő információ helyes (sajnos az indoklás már pontatlan):a válasz helyes

Magyarázat a megoldáshoz

4. feladat 0 / 5 pont

Nap végi vitatéma a sör mellett mielőtt hazamennétek. Melyik népszerű parancssori eszközzel lehet a legkönnyebben (legkevesebb karakterből álló utasítással/utasításokkal) kiszámolni a Pi számot 10000 számjegyig?

- [ ] awk
- [x] bc
- [ ] grep
- [ ] sed
- [ ] vi

Magyarázat a megoldáshoz

A bc (basic calculator) pont arra való, hogy tetszőleges pontossággal számoljunk, de örömmel várjuk a kreatív ötleteket, hogy hogyan lehetne a többivel ugyanezt hatékonyabban megtenni.

Vissza a kategóriáimhoz

Linux rendszerfejlesztés és üzemeltetés (One Identity)
2. forduló

Ismertető a feladathoz

A kezdeti izgalmak elmúltával következnek a hétköznapi kihívások.
Lássuk mivel tölti nálunk az átlagos napokat egy Linux guru!

Tekintettel arra, hogy egy választ sem rögzítettél az alábbi feladatlapon, ebben a fordulóban a kitöltésére rendelkezésre álló idő teljes egésze, azaz 10 perc került rögzítésre mint megoldáshoz felhasznált idő.

1. feladat 0 / 5 pont

A free parancs alábbi kimenetei közül melyik esetben van elég szabad kapacitás ahhoz, hogy egy 1GB memóriaigényű alkalmazást indítsunk a swap igénybevétele nélkül?

- [ ]
```
endre@home:~$ free -g
              total        used        free      shared  buff/cache   available
Mem:              7           6           0           1           1           0
Swap:             9           0           8
```
- [x]
```
endre@home:~$ free -m
              total        used        free      shared  buff/cache   available
Mem:           7806        4168        2146        1061        1471        2377
Swap:          9999         816        9183
```
- [x]
```
endre@home:~$ free -h
              total        used        free      shared  buff/cache   available
Mem:           7.7G        2.0G        135M        1.0G        5.6G        4.4G
Swap:          9.8G        539M        9.2G
```
- [ ]
```
endre@home:~$ free -m
              total        used        free      shared  buff/cache   available
Mem:           7806        6418         159        1060        1309         145
Swap:          9999         979        9020
```
- [ ]
```
endre@home:~$ free
              total        used        free      shared  buff/cache   available
Mem:        8074028     6023536      129392     1087908     1923100      670132
Swap:      10239996      756736     9483260
```

Magyarázat a megoldáshoz

Egyedül az Available értéket kell nézni. Ahol ez nagyobb mint 1GB, ott van lehetőség az alkalmazás elindítására swap nélkül.

Kisebb nehezítés, hogy a free különböző egységekben írja ki a válaszokban az eredményt, ehhez a free útmutatóját érdemes elővenni.

2. feladat 0 / 5 pont

Szeretnéd a HTTPS portot megnyitni külső kapcsolatok számára iptables segítségével. Internetes források az alábbi parancsot javasolják, melynek eredménye is alább látható. Kívülről a HTTPS port továbbra sem elérhető. Mi lehet a gond?

```
iptables -A INPUT -p tcp --dport 443 -m conntrack --ctstate NEW,ESTABLIS
```
```
[root@fedoraa ~]# sudo iptables -L
Chain INPUT (policy ACCEPT)
target     prot opt source               destination
ACCEPT     all  --  anywhere             anywhere             state RELATED,ESTABLISHED
ACCEPT     icmp --  anywhere             anywhere
ACCEPT     all  --  anywhere             anywhere
ACCEPT     tcp  --  anywhere             anywhere             tcp dpt:smtp
ACCEPT     tcp  --  anywhere             anywhere             tcp dpt:http
ACCEPT     tcp  --  anywhere             anywhere             state NEW tcp dpt:ssh
REJECT     all  --  anywhere             anywhere             reject-with icmp-host-prohibited
ACCEPT     tcp  --  anywhere             anywhere             tcp dpt:https ctstate NEW,ESTABLISHED

Chain FORWARD (policy ACCEPT)
target     prot opt source               destination
REJECT     all  --  anywhere             anywhere             reject-with icmp-host-prohibited

Chain OUTPUT (policy ACCEPT)
target     prot opt source               destination
```

- (•) Egy meglévő REJECT előbb kerül kiértékelésre mint az új ACCEPT
- ( ) A kapcsolódó OUTPUT szabályt is hozzá kell adni
- ( ) A kapcsolódó FORWARD szabályt is hozzá kell adni
- ( ) A tűzfal szabály megfelelő, valamilyen más probléma lehet (pl. route tábla)
- ( ) Felesleges, mert a policy ACCEPT

Magyarázat a megoldáshoz

Próbáld ki! :)

3. feladat 0 / 5 pont

Az egyik saját fejlesztésű alkalmazásunk időnként leáll és a naplófájlban "Too many open files" hibákat találunk. Ez nagyon furcsa, mert az alkalmazás egyszerre legfeljebb két fájlt olvas/ír a fájlrendszerből. Mi lehet a probléma?

- ( ) Operációs rendszer szintjén nincsenek ilyen korlátozások, az alkalmazásban lehet valamilyen limitáció.
- ( ) Ez csak akkor lehetséges ha az alkalmazás nem zárja be megfelelően a fájlokat.
- (•) Linux esetében a nyitott hálózati kapcsolatok is fájlleírókat használnak."ulimit -n" segítségével ellenőrizni lehet az "open files" limitet.
- ( ) A hiba arra utal, hogy már nincs elég memória szabadon, így nincs lehetőség új fájlok megnyitására.

Magyarázat a megoldáshoz

4. feladat 0 / 4 pont

Mivel lehet kideríteni, hogy az előző feladatban szereplő alkalmazás mennyi fájlt tart nyitva?

- [x] ls
- [x] eBPF tools
- [ ] man
- [ ] ulimit
- [x] lsof
- [ ] netstat
- [ ] ipables
- [ ] a fentiek közül egyik sem

Magyarázat a megoldáshoz

5. feladat 0 / 5 pont

Melyik paranccsal törölhető a --cica nevű fájl?

- ( )
```
rm \-\-cica
```
- ( )
```
rm "--cica"
```
- (•)
```
rm -- --cica
```
- ( )
```
rm --cica
```
- ( )
```
rm '--cica'
```

Magyarázat a megoldáshoz

Vissza a kategóriáimhoz

Linux rendszerfejlesztés és üzemeltetés (One Identity)
3. forduló

Ismertető a feladathoz

A március váratlan kihívásokkal köszöntött be. Az irodát bezárták, otthonról próbált mindenki szembenézni a váratlan új feladatokkal.
A rendszereinket ráadásul sokkal többen használják, a szokásos terhelés többszöröse érkezik. A vezetőség költségcsökkentésbe kezdett, nem vehetünk jobb vasat.
Meg tudod oldani a lehetetlent?

Tekintettel arra, hogy egy választ sem rögzítettél az alábbi feladatlapon, ebben a fordulóban a kitöltésére rendelkezésre álló idő teljes egésze, azaz 10 perc került rögzítésre mint megoldáshoz felhasznált idő.

1. feladat 0 / 5 pont

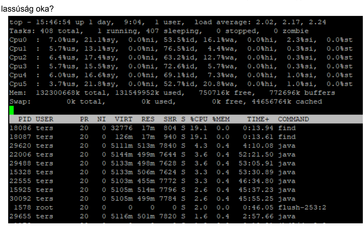
Egyre gyakrabban fordul elő, hogy megszakadnak az SSH kapcsolatok az otthoni géped és a Linux szerver között. Az irodai munka során ez ritkán okozott gondot, de most kötelező otthonról dolgozni és megoldást kell találnod erre a problémára, hogy hatékonyabban tudj dolgozni.
Melyik tanács segíthet?

- [x] A screen használatával folytatni lehet a munkát a kapcsolat megszakadása után is, amíg ki nem derítjük, mi az oka a szakadásnak, és azt orvosoljuk
- [ ] A ClientAliveInterval szerver oldali beállításával megengedhetjük, hogy egy bizonyos ideig újra lehessen csatlakozni a megszakadt SSH folyamatokhoz. A legtöbb kliens ezt automatikusan meg is teszi.
- [ ] A ServerAliveInterval szerver oldali beállításával megengedhetjük, hogy egy bizonyos ideig újra lehessen csatlakozni a megszakadt SSH folyamatokhoz. A legtöbb kliens ezt automatikusan meg is teszi.
- [ ] Érdemes a telnet protokollt használni, mert az UDP alapú, így nem érinti a TCP kapcsolatok megszakadása

Magyarázat a megoldáshoz

2. feladat 0 / 5 pont

A felhasználók panaszkodnak a lassú rendszer miatt. A mellékelt top kimenet alapján mi lehet a lassúság oka?



- [ ] A load average alapján a szerver több mint kétszeresen túl van terhelve. Több processzor hozzáadása szükséges.
- [ ] A rendelkezésre álló összmemóriához (132 300 668 k) képest nagyon kevés a szabad memória (750 716 k). Több memória hozzáadásával lényegesen gyorsulhat a rendszer.
- [ ] A processzorok száma (6 darab) arra utal, hogy elavult architektúrára épül a rendszer (2015 előtti). Hasonló kapacitású, de új szerver lényegesen gyorsabb lenne.
- [x] A processzorok magas %wa kihasználtsága arra utal, hogy sokat vár I/O műveletekre a rendszer. Érdemes lehet egyéb eszközzel (pl. iotop) tovább vizsgálódni.
- [ ] A szerver sok java alapú alkalmazást futtat. A java alkalmazások köztudottan lelassítják a szervereket. Ezeket le kell állítani.
- [ ] A szervernek bőven van rendelkezésre álló kapacitása, máshol kell keresni a problémát.

Magyarázat a megoldáshoz

3. feladat 0 / 5 pont

A PHP alapú weboldalunk Apache 2.4.46 webszerveren fut. Nagy terhelés alatt azt vesszük észre, hogy:
- a felhasználóknak nagyon lassan vagy egyáltalán nem töltődik be a weboldal,
- a memória és a swap teljes kihasználtságon van,
- véletlenszerűen leállnak apache processzek.

Általában csak újraindítás után áll helyre a rendszer. Az alábbiak közül melyik lehet a probléma oka és hogyan lehetne megoldani?

- [ ] A swap elfogyása miatt lassul be a rendszer, növelni kell a rendelkezésre álló swap méretét.
- [ ] Memóriaszivárgás (memory leak) van a HTML kódban. Át kell nézni a kódot.
- [ ] Az apache szerver köztudottan nem képes nagy terhelést kiszolgálni. Nagy terhelésre javasolt saját szervert írni.
- [x] Túl sok kérést próbál párhuzamosan kiszolgálni az apache. Érdemes lenne állítani az mpm prefork modul beállításain (pl. MaxRequestWorkers).
- [ ] A feldolgozásra váró kapcsolatok sok memóriát foglalnak. Limitálni kell a várakozó kapcsolatok számát az mpm common direktívák segítségével (pl. ListenBackLog).
- [ ] Lassú a PHP weboldal alkalmazás, hívni kell a fejlesztőket.

Magyarázat a megoldáshoz

Túl sok feldolgozó process túl sok memóriát fogyaszt. A feldolgozó processzek számát a rendelkezésre álló memória függvényében kell beállítani.

4. feladat 0 / 5 pont

Az egyik legnagyobb terhelést kiszolgáló Linux szerverünk a vállalati követelményeknek megfelelően az Active Directory (AD) segítségével authentikálja a felhasználókat. Ennek a megvalósításához a System Security Services Daemon (SSSD 1.16.5) lett beállítva az alább megadott legfontosabb paraméterekkel. A felhasználók panaszkodnak, hogy nagyon lassú a bejelentkezés (több mint 30 másodpercig tart).
Melyik SSSD változtatás javíthat érdemben a helyzeten a dokumentáció alapján?

```
auth_provider = krb5
access_provider = ldap
chpass_provider = krb5
id_provider = ldap
sudo_provider = ldap
```

- [ ] "enumerate = true" sor beszúrása
- [ ] "ignore_group_members = false" sor beszúrása
- [ ] krb5_auth_timeout növelése
- [ ] krb5_auth_timeout csökkentése
- [x] "cached_auth_timeout = 120" sor beszúrása

Magyarázat a megoldáshoz

A cached_auth_timeout engedélyezi, hogy lokálisan cache alapján azonosítsunk egy felhasználót.

Vissza a kategóriáimhoz

Linux rendszerfejlesztés és üzemeltetés (One Identity)
4. forduló

Ismertető a feladathoz

2020. Május
A csapatod nagyon sikeres volt az elmúlt hónapokban a nehéz körülmények ellenére. Ti fogtok kiépíteni egy új rendszert, mely kritikus üzleti fájlokat fog megosztani ügyfelekkel.
A csapat most készíti elő a terveket. Tudsz segíteni?

Tekintettel arra, hogy egy választ sem rögzítettél az alábbi feladatlapon, ebben a fordulóban a kitöltésére rendelkezésre álló idő teljes egésze, azaz 10 perc került rögzítésre mint megoldáshoz felhasznált idő.

1. feladat 0 / 5 pont

Szeretnénk fájlokat több felhasználó mappájában is elérhetővé tenni. A csapatban vita alakult ki arról, hogy hardlinket vagy symbolic linket használjunk. Válasszuk ki a hasznos megjegyzéseket.

- [ ] Hardlink segítségével mappákat is lehet hivatkozni
- [ ] Hardlink esetében mindig több tárhelyre van szükség
- [x] Hardlink segítségével megoldható, hogy a fájl tartalmának módosítása az összes felhasználónál egyszerre látszódjon
- [x] Symbolic link segítségével különböző NFS hálózati megosztások között is lehet hivatkozni
- [ ] Symbolic link lényegesen gyorsabb mint a hardlink (a fast symlink szabvány használatával)
- [ ] Symbolic link használata esetén a hivatkozott fájl jogosultságaitól függetlenül a fájl tartalma mindig olvasható lesz (feltételezve, hogy a symbolic link olvasható)
- [ ] Biztonsággal és teljesítménnyel kapcsolatos problémák miatt egyik használata sem javasolt

Magyarázat a megoldáshoz

2. feladat 0 / 5 pont

SFTP szervert üzemeltetünk ProFTPD 1.3.7a segítségével. Biztonsági okokból a felhasználók be vannak zárva chroot segítségével a saját könyvtárukba (DefaultRoot ~). Hogyan tudunk fájlokat vagy mappákat elérhetővé tenni a felhasználók számára úgy, hogy olvashatóak legyenek még akkor is ha a saját könyvtárukon kívül vannak?

- ( ) Ez nem lehetséges, pontosan ez a chroot célja
- ( ) Softlinket kell létrehozni, nem szükséges más beállítás
- (•) mount --bind használatával
- ( ) Softlinket hozunk létre, de ezt külön engedélyezni kell a ProFTPD konfigurációjában (AllowChrootSymlinks on)

Magyarázat a megoldáshoz

3. feladat 0 / 5 pont

Biztonságos bejelentkezéshez SSH kulcs alapú authentikációt állítottál be. Mindent a leírás szerint csináltál, de nem akar működni a kulcs alapú bejelentkezés, továbbra is csak jelszóval lehet belépni. Az alábbiak közül melyik okozhatja azt, hogy a kulcs alapú bejelentkezés elutasításra kerül?

- ( ) A privát kulcs időbélyege túl régi. Biztonsági okokból a Linux operációs rendszerek maximum két éves kulccsal engedik a bejelentkezést.
- ( ) Az RSA kulcs használata számításigényes, ezért csak többprocesszoros gépen működik
- ( ) A kulcs alapú authentikáció csak akkor működik ha biztonsági okokból letiltjuk a jelszó alapú bejelentkezést
- ( ) A kulcspár nem azon a gépen lett generálva ahol használni szeretnénk
- ( ) A ~/.ssh/id_rsa fájl hiánya a szerveren
- ( ) Az ~/.ssh/authorized_keys fájl hiánya a kliensen
- (•) Túl megengedő jogosultságok a ~/.ssh/authorized_keys fájlon

Magyarázat a megoldáshoz

Vissza a kategóriáimhoz

Linux rendszerfejlesztés és üzemeltetés (One Identity)
5. forduló

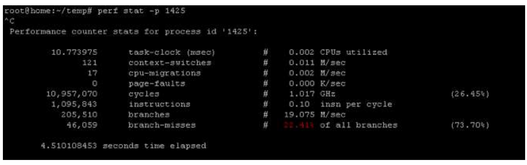
Ismertető a feladathoz

2020. július
Végre itt a nyár! Mindenki szabadságon, a Linux csapat csak ügyeletet tart, nem fogunk bele új fejlesztésbe. Mégsem telik eseménytelenül a nyár, érdekes kérdéseket kaptunk az alkalmazásfejlesztő csapatoktól. További izgalom, hogy váratlanul új virtuális gépekkel kell bővítenünk kapacitásainkat, mivel megnövekedett az érdeklődés az egyik termékünk iránt.
Sok sikert!

Tekintettel arra, hogy egy választ sem rögzítettél az alábbi feladatlapon, ebben a fordulóban a kitöltésére rendelkezésre álló idő teljes egésze, azaz 30 perc került rögzítésre mint megoldáshoz felhasznált idő.

1. feladat 0 / 10 pont

Az alábbi perf statisztikák között mit jelent a branch-misses sor?



- A program "if" utasításai közül hány esetben ugrottunk az "else" ágra.
- A program "if" utasításai közül hány esetben ugrottunk a "then" ágra.
- Az operációs rendszer hány esetben becsülte meg tévesen, hogy az elágazáson merre halad tovább a program futása
- A processzor hány esetben becsülte meg tévesen, hogy az elágazáson merre halad tovább a program futása
- Az utasítás ellenőrző kódja (command checksum) hány esetben jelzett hibát
- A memória ellenőrző kódja (memory checksum) hány esetben jelzett hibát

Magyarázat a megoldáshoz

2. feladat 0 / 20 pont

A megemelkedett terhelést csak úgy tudjuk kezelni, hogy az eddigi fizikai gépeink mellé új virtuális gépeket bérlünk. Az új gépeken az egyik alkalmazásunk sokkal lassabb mint a régin.
A mellékelt strace kimenet alapján melyik tényező befolyásolja a legjobban a teljesítményt?

- A dev / random nem képes megfelelő sebességgel véletlen számokat generálni
- A hiányzó fájlok keresése sok időt vesz igénybe (No such file or directory)
- A hibás útvonalak használata (/var//libguestlib.so.0.0.0) konfigurációs problémára utal, mely lassítja az alkalmazást
- Az alkalmazás nem lett optimalizálva az új környezetre. Érdemes lenne újrafordítani egy modern 64 bites processzoron.
- Az strace kimenet nem alkalmas arra, hogy ezt megállapítsuk, mivel csak az operációs rendszerrel való interakciókat tartalmazza. Az időbélyegek között az alkalmazás egyéb műveletek végezhet.

Magyarázat a megoldáshoz

A random hívások időnként több mint 5 másodpercet igényelnek:

getrandom("\xe9\x4b\x55\x86\x11\xfd", 100, GRND_RANDOM) = 6 <5.166308>

3. feladat 0 / 10 pont

Az egyik alkalmazás váratlan időpontokban 100% körüli processzor kihasználtsággal kezd futni, amíg újra nem indítjuk. Az alkalmazás fejlesztőinek nagy segítség lenne, ha meg tudnánk mondani melyik szál (thread) okozza ezt.
Hogyan lehetne ezt megállapítani?

- általánosságban ez nem megállapítható, csak az adott technológiára jellemző eszközökkel (pl. C++ esetén gprof, Java esetén JConsole)
- nem lehet megállapítani
- Linuxon minden szál külön folyamatban fut, ez a kérdés csak Windowson értelmezhető
- top segítségével a szálakat is lehet monitorozni

Magyarázat a megoldáshoz

Vissza a kategóriáimhoz

Linux rendszerfejlesztés és üzemeltetés (One Identity)
6. forduló

Ismertető a feladathoz

2020. szeptember
Mindig is szeptember volt a kedvenc hónapod. Minden évben ilyenkor ültettél egy fát, ráadásul a kislányod is szeptemberben született. Idén ennek ellenére az aggodalom motoszkált benned, nehéz éved volt és még nincs vége. Át akarod gondolni, hogy megfelelően védve vannak-e a szerverek egy esetleges támadással szemben. Sikerül minden kétségednek a végére járni?

Tekintettel arra, hogy egy választ sem rögzítettél az alábbi feladatlapon, ebben a fordulóban a kitöltésére rendelkezésre álló idő teljes egésze, azaz 30 perc került rögzítésre mint megoldáshoz felhasznált idő.

1. feladat 0 / 10 pont

Szabályozói követelmények miatt a teljes /var/log mappáról minden nap biztonsági mentést kell készítenünk. Azt tapasztaltuk, hogy ezek a mentések a vártnál sokkal több helyet foglalnak. Az ls parancs szerint a lastlog fájl a legnagyobb:

```
[root@vm log]# ls -lh lastlog
-rw-r--r-- 1 root root 354G Jul 30 09:29 lastlog
```

Mi okozhatja ezt?

- A lastlog minden felhasználóról eltárolja, hogy mikor jelentkezett be utoljára. Minden bizonnyal nagyon sok felhasználó használja a rendszert.
- A lastlog tömörített formában tárol adatot több más logfájlból. Egy másik fájl nagy mérete okozza a problémát.
- Minden bizonnyal debug szinten kerülnek naplózásra a bejelentkezések. Ez nem javasolt éles rendszerekben.
- **A lastlog fájlnak csak látszólag ilyen nagy a mérete, a legtöbb fájlrendszer hatékonyan el tudja tárolni sokkal kisebb tárhelyen**

Magyarázat a megoldáshoz

A lastlog sparse fájl: https://hu.wikipedia.org/wiki/Ritka_f%C3%A1jl

2. feladat 0 / 10 pont

A Dirty COW (https://en.wikipedia.org/wiki/Dirty_COW) talán az egyik legismertebb sebezhetőség a Linux Kernel történetében. Maga Linus Torvalds javította ezt a sebezhetőséget 2016 október 18-án. Milyen git parancsokkal tudjuk megnézni hogyan javították ezt?

- git clone https://git.kernel.org/pub/scm/linux/kernel/git/stable/linux-stable.git
  git diff 18.10.2016
- **git clone git://git.kernel.org/pub/scm/linux/kernel/git/stable/linux-stable.git**
  git show 19be0eaffa3
- git clone git://git.kernel.org/pub/scm/git/git.git
  git diff 19be0eaffa3
- git clone https://git.kernel.org/pub/scm/git/git.git
  git show mm/gup.c
- git co https://git.kernel.org/pub/scm/git/git.git
  git show mm/gup.c
- git co git://git.kernel.org/pub/scm/linux/kernel/git/stable/linux-stable.git
  git commit 18.10.2016

Magyarázat a megoldáshoz

A wikipedia szócikk hivatkozza az adott commitot:

https://git.kernel.org/pub/scm/linux/kernel/git/torvalds/linux.git/commit/?id=19be0eaffa3ac7d8eb6784ad9bdbc7d67ed8e619

3. feladat 0 / 20 pont

A csatolt syslog kimenet alapján hány különböző felhasználó lépett be sikeresen ssh kapcsolaton keresztül?

A megoldások:
9
9 (curi, fztu, hxu, jmzhu, suyuxin, tzhao, xxchen, yuewang, zachary)

Magyarázat a megoldáshoz

A "session opened for user" sorokat kell kigyűjteni és megszámolni az egyedi felhasználóneveket.

Vissza a kategóriáimhoz

Linux rendszerfejlesztés és üzemeltetés (One Identity)
7. forduló

Ismertető a feladathoz

2020. December
Váratlan meglepetést tartogatott még 2020, valaki megtámadta a kedvenc Linux szervered!
Meg tudod fejteni, hogy mi történt?

Tekintettel arra, hogy egy választ sem rögzítettél az alábbi feladatlapon, ebben a fordulóban a kitöltésére rendelkezésre álló idő teljes egésze, azaz 30 perc került rögzítésre mint megoldáshoz felhasznált idő.

1. feladat 0 / 10 pont

Támadás érte a Linux szerverünket, de szerencsére csak egy vicces szöveges fájlt hagyott a támadó a root user home könyvtárában.
A szerveren a sudo 1.8.21p2-as verziója fut.
Feltételezve, hogy nincs más sérülékenység a rendszerben, melyik sudo szabályt használhatta ki egy felhasználó ahhoz, hogy root jogokkal írhasson egy fájlt?

- [ ]
```
ALL ALL= /sbin/poweroff
```
- [ ]
```
ALL ALL= (root) NOEXEC: /bin/cat /var/log/*
```
- [x]
```
ALL ALL= (root) /bin/less /var/log/messages
```
- [ ]
```
ALL ALL= (root) /usr/bin/su www
```
- [ ]
```
root ALL= (ALL) ALL
```
- [x]
```
ALL ALL= (root) /bin/less /var/log/*
```
- [ ]
```
root ALL= (root) /usr/bin/vi
```

Magyarázat a megoldáshoz

A less (NOEXEC hiányában) lehetővé teszi, hogy a futtatás során egy parancsokat hajtsunk végre.

A többi szabály vagy csak a root felhasználó számára ad hozzáférést vagy önmagában nem elegendő fájlok írásához.

2. feladat 0 / 10 pont

A root user home mappájában (a hacked.txt fájlban) az alábbi üzenetet találtuk. Hogyan léphetünk kapcsolatba H4B17-tel?

Szia!
Ha érdekel, hogy hogyan jutottam be a szerveredre, csatlakozz a marmegintegysebezhetolinuxszerver.hu 11464 portjára TCP kapcsolaton nc-vel. Én is csatlakozni fogok hozzád ugyanezen a porton egy külön TCP kapcsolaton.
H4B17

A szerver netcat verziója: Debian patchlevel 1.187-1ubuntu0.1

- ( )
```
nc -d marmegintegysebezhetolinuxszerver.hu -p 11464
nc -l -Oport=11464
```
- ( )
```
nc -n marmegintegysebezhetolinuxszerver.hu -p 11464 -P TCP
nc -listen -port 11464
```
- (•)
```
nc marmegintegysebezhetolinuxszerver.hu 11464
nc -l -p 11464
```
- ( )
```
nc -u marmegintegysebezhetolinuxszerver.hu -p 11464
nc -l -p 11464
```
- ( ) Ez csak egy vicc, nem lehet megoldani, mert a port csak egyetlen egy TCP kapcsolatot tud kezelni.

Magyarázat a megoldáshoz

Csak ez a válaszlehetőség felel meg az nc használatának.

3. feladat 0 / 20 pont

H4B17 leírta, hogy hogyan jutott be a szerverünkre, de ezt az információt XOR titkosítással (https://en.wikipedia.org/wiki/XOR_cipher) kódolta. Találtál egy python alapú eszközt (https://github.com/hellman/xortool) amivel talán meg lehetne fejteni.
Milyen kulcsszóval lett kódolva a csatolt zipben található fájl?

A megoldások:
Habit
Habi5
hABIT
habit
Nem az jut legmesszebbre az úton, aki a leggyorsabb, vagy a legszívósabb, hanem aki nem hagyja abba a járást. Ha megállsz, vesztettél. De mindaddig, amíg lépkedsz, van esélyed célba érni. És közben elhagyod mindazokat, akik megálltak, legyenek ők bármennyivel is jobbak nálad.

Magyarázat a megoldáshoz

Vissza a kategóriáimhoz