8 gyakori Unix™ parancssori hiba, amit a felhasználók elkövetnek

Még a legjobbak is hibáznak. Bemutatunk néhányat a leggyakoribb Unix™ parancssori hibák közül.

COLLABORATORS			
	TITLE : 8 gyakori Unix™ parancssori hiba, amit a felhasználók elkövetnek		
ACTION	*NAME*	*DATE*	*SIGNATURE*
WRITTEN BY		2014. február 2.	

Tartalomjegyzék

A példák listája

Kivonat

Eredeti cikk: 8 Common Unix Command Line Mistakes That Users Make

2014. január 8., szerda: A **Unix**™ felhasználók jó néhány parancsot memorizálnak és óvatosan használnak. Néha mégis hibák történnek. Íme néhány a leggyakoribb hibákból, amelyeket elkövethetünk és el kellene kerülni.

> Nem tudtam megállni, hogy az egy-egy mondatos ismertető mellett néhány megjegyzést ne tegyek. Az eredeti anyagban ugyanis a szűkszavúság az érthetőség rovására ment – szerintem.
>
> Megjegyzéseim, ehhez hasonlóan, beljebb helyezkednek el, így nyugodtan át lehet lépni, ha valakit nem érdekelne.

1. userdel parancs

Véletlenül véglegesen eltávolíthatod az egész felhasználói fiókot ennek a parancsnak a használatával.

> Kétségtelenül igen veszélyes a parancs, de ha csak a *login* (felhasználó) nevet adjuk meg, akkor számos helyről a hivatkozást a parancs kitörli, de a belépési könyvtár, annak tartalma megmarad, vagyis még esély lehet a hiba következményeinek legalább részleges felszámolására. De ha a `-r` opciót is megadjuk, akkor okozunk különösen nagy bajt. Hátha még a `-f` opciót is megadjuk (ami még az ellenőrző kérdést is elnyomja...)

2. Solaris rendszer újratöltése

Linuxon a `killall` paranccsal név alapján lehet egy proceszt megszakítani. Solarison ez a parancs minden akciót megszakít.

> Ez az eset is rávilágít arra, hogy egyik Unix változaton működő parancs a másikon nem ugyanúgy működhet. Éppen ezért *mindig utána kell nézni adott parancs működésének, ha másik rendszeren dolgozunk*!

3. Tönkretett `named.conf`

Ez a **./mkzone example.com > /var/named/chroot/etc/named.conf** futtatásával okozható.

> Ez a parancs névkiszolgáló zónafájlok létrehozására szolgál. A jelzett parancs az aktív névkiszolgáló beállításához/karbantartásához használatos. Véleményem szerint az a felhasználó, aki ilyet üzemeltet, általában tisztában van ezekkel a veszélyekkel, de ha felhívták erre a figyelmet, akkor nyilván volt rá példa...
>
> A parancs egy szkript. Rövid bemutatása, illetve maga a szkript megtekinthető: Shell Script To Create BIND Zone Files.

4. `tar`-ral és `rsync`-kel tönkretehető a biztonsági mentés

Ezeknél a parancsoknál a `-x` opció helyett a `-c` opció használata okozhatja ezt a problémát.

> Mindkét parancs azt feltételezi, hogy a felhasználó tudja, mit csinál, ezért nem kérdez, hanem végrehajt. Szerencsére a két opció eltérő szintaxist igényel:

Example 4.1 A `tar` kétféle használatának összehasonlítása

```
tar cf <létrehozandó archívum> <archiválandó fájl(ok)>

tar xf <kibontandó archivum> [<kibontandó fájl(ok)>]
```

A második esetben a kibontandó fájlok listája nem kötelező paraméter, ezért ha azt nem adjuk meg, de az említett hibát elkövetjük, hibaüzenetet kapunk, de aparancsot a rendszer nem hajtja végre. Viszont akár csak egy-egy fájlt akarunk kibontani, ezért második paraméter(eke)t is adunk hibás opció használattal, akkor megtörténik a baj! Éppen ezért

Megjegyzendő!
Mindig olvassuk el figyelmesen a beütött parancsot, mielőtt a <Enter>-t megnyomjuk!

5. Kitörölt Apache `DocumentRoot`

Ez bekövetkezhet a http mappán kiadott `rm -rf` paranccsal.

Nyilvánvalóan nem arra hívja fel a figyelmet, hogy a felhasználó képes lenne a WEB szerver teljes tartalmát kitörölni, ehhez ugyanis általában `root` jogok, vagy a WEB szerver futtatására jogosult felhasználói (alapértelmezetten Debianon `www-data`, **PCLinuxOS**-en `apache`) hozzáférésre van szükség. Ezzel szemben általában az egyes felhasználók saját maguk is készíthetnek alapértelmezett mappájukban olyan almappát (alapértelmezetten `public_html` néven), ami a WEB szerveren keresztül elérhető. Ez aztán a honlapon keresztül látható is (alapértelmezetten ~<felhasználónév> almappában, pl.: `http://localhost/~janu`). Ennek a mappának a véletlen kitörlése történhet így.

Megszívlelendő!
A legveszélyesebb parancsok közül is kiemelkedik a `rm` (más rendszereken `del[ete]`) parancs. Ennek a `-f` opcióját lehetőség szerint kerüljük, de ha használjuk, akkor **különösen óvatosan** járjunk el!

6. Véletlenül megváltoztatott gazdanév, ami hamis riasztást vált ki

Ha megváltoztatod géped gazdanevét egy fürt (cluster) csomópontéra, ez figyelmeztető üzenetet okozhat.

Nem ritka, hogy szerverek, munkaállomások fürtökbe vannak kötve, ez gazdaságos és hatékony megoldás lehet. Az egyediség biztosítására a „cluster software” általában egy generált egyedi névvel azonosítja az adott csomópontot (vagy valamilyen szabály alapján rögzített nevet ad). Gondoljunk csak arra, hogy hány `localhost.localdomain` lehet a világban :-) Az új gazdanév nem gondos átvezetése riasztást adhat.

Valódi riasztás
Számos olyan – elsősorban hálózatos – program létezik, amely a gép gazdanevéhez kötött (pl. számos grafikus felület foglalatokat (socket) hoz létre, aminek a nevét a gazdanév alapján képzi). A gazdanév végleges megváltoztatását követően mindig célszerű újraindítani a gépet, mert az eltérés működési elégedetlenséghez vezethet!

7. A publikus hálózat interfészének leállítása

Esetenként a VPN (virtuális magán hálózat) interfészének leállítása a publikus hálózati interfész leállását is okozhatja.

A VPN interfész nem egy önálló eszköz, hanem a publikus hálózati interfész alias-a. Ilyen lehet például a tap<x> vagy tun<x> interfész. Egyes esetekben ennek a leállítása okozhatja a fizikailag létező interfész (eth<x>, wlan<x>) leállását is.

megjegyzés
Ezt a cikket a hibajelzések alapján állították össze, tehát előfordulhatott ilyen eset. Én még nem találkoztam ezzel a problémával.

8. Tűzfal zárlat

Az `sshd_config`-ban a port(ok) megváltoztatása tűzfal zárlathoz vezethet.

A szerverek karbantartását ma már jellemzően távolról végzik. Ehhez a biztonságosabb `ssh` bejelentkezést használják, bár egyre több más megoldás is napvilágot látott (ezek közül az egyik legrégebbi alkalmazás a *webmin*). Ha az `ssh` kommunikációs portját megváltoztatjuk, de előzőleg a tűzfalon az új portot nem engedélyeztük, az *sshd* újraindításával kizárjuk magunkat a rendszerből.

megjegyzés Ha nincs fizikai hozzáférésünk a szerverhez, ez bonyodalmat okoz. Némi segítséget jelenthet alternatív alkalmazás megléte (erre jó a *webmin*), amelyen keresztül hibánkat korrigálhatjuk.