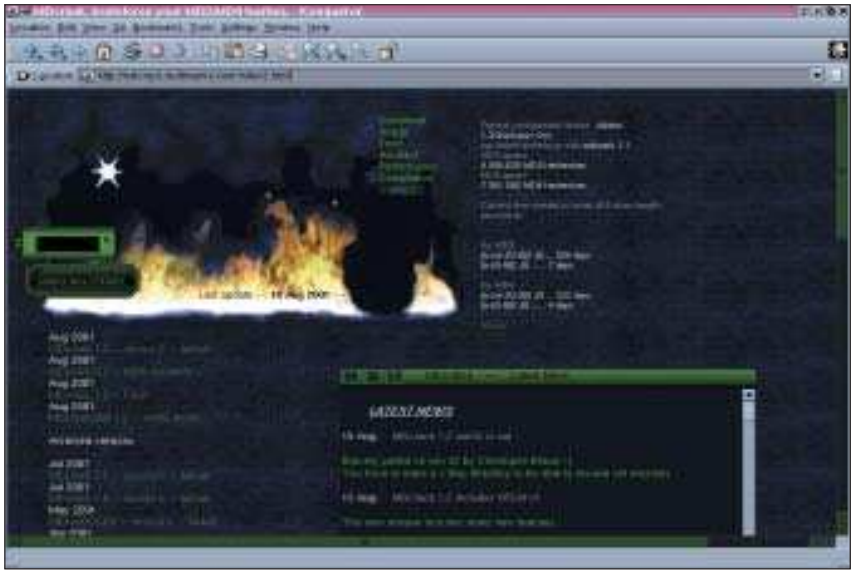
A jelszavas védelem fejlődése

Túlságosan kiszolgáltatott lenne a rendszerünk? Bemutatjuk hogyan hozhatjuk ki a legtöbbet az olyan Unix-alapú jelszavas védelmi rendszerekből, mint az MD5 és a PAM.

Ha látni szeretnéd, hogyan működik az egyedfejlődés, csak vess egy pillantást a GNU/Linux jelszavas rendszereinek fejlődésére! Bár az alapállapotban minden Unix-rendszerben meglévő jelszavas védelem már csak csökevényes szerve a rendszernek, a számítógépes betörők alakjában fellépő természetes kiválasztódás nyomására olyan fejlettebb védelmi rendszereket indítottak el a fejlődés útján, mint az árnyékjelszavak (shadow password), az MD5- és a PAM-módszer. Azóta a jelszavas védelem olyan új területeken is meghonosodott, mint például az indításkezelők, távoli bejelentkezések és egyéb fejlett biztonsági rendszerek. Nap mint nap használjuk ezeket a védelmi eszközöket, de ha a rendszerünket biztonságosabbá szeretnénk tenni, alaposabban meg kell ismerkedni velük. A Unix mára már szabványossá vált jelszavas rendszere a védelmi rendszerek fejlődésének szempontjából jura korszaknak számító 70-es években indult el. Eltekintve attól, hogy manapság a legtöbben grafikus felület alól, például a `gdm` program segítségével használják, az idők folyamán szerkezete nem sokat változott.
A folyamat maradt a régi: a rendszerbe való bejelentkezéskor a felhasználó egy legfeljebb nyolc (korábban hat vagy hét) karakterből álló jelszót ad meg, amelyet azután a DES (adattitkosítási szabvány) algoritmus egy titkos kulcsba kódol. Ez a kulcs a továbbiakban a */etc/passwd* állomány második oszlopába kerül, ahol tulajdonképpen bárki hozzáférhet.
Az eltelt idő során azonban a jelszavas védelmi rendszerek mégis megváltoztak és fokozódott a számítógépes betörők közti versengés is. A DES-algoritmus ma már másodpercek alatt feltörhető. Még elkeserítőbbé teszi a helyzetet, hogy a hagyományos jelszavas rendszer a kulcsokat mindenki által hozzáférhető, nyilvános helyeken tárolja, ahol a behatolók könnyűszerrel rátalálhatnak.
Az egyetlen megoldás az lehetne, ha szigorú korlátozásokkal sújtanánk a felhasználókat, és még az olyan gyakori parancsok kiadását is letiltanánk, mint az `ls -l`. Bár néhány biztonsággal foglalkozó szakember akkor lenne igazán boldog, ha a megvédendő gépet kikapcsolva, több kilométer mélységben lévő ólomfalú pincében helyezhetnék el, be kell látnunk, hogy az ilyen kemény megszorítások nem vezetnek megoldásra.
A '90-es évek közepén a már széles körben használt Internet egyre több lehetőséget kínált a betörőknek még megfeszítettebbé téve a versengést. Válaszlépésként különböző védelmi rendszerek indultak fejlődésnek. (Ehhez kapcsolódik a 46. oldalon lévő Könnyű álmok című cikkünk.) Kezdetben ezeket az eszközöket csak utólagos kiegészítőként lehetett a rendszerbe illeszteni, a kilencvenes évek második felében beköszöntő „Új kor hajnalán” viszont már egymást segítve fejlődtek, és a Linux-terjesztések alapcsomagjaiba is bekerültek.

Az árnyék válasza

Az árnyékjelszavak a nevüket onnan kapták, hogy ők a hagyományos jelszavak rejtett megfelelői. A különbség az, hogy az árnyékjelszavak lakhelye nem a mindenki számára hozzáférhető /etc/passwd, hanem a */etc/shadow* állomány második oszlopa, amelyet csak a rendszergazda képes „birtokolni”. Ebben az oszlopban Linux-változattól függően egy csillag vagy egy felkiáltójel mutatja, ha az adott felhasználóhoz nincs jelszó beállítva. A /etc/passwd jelszó oszlopában mindössze egy *x* utal az árnyékjelszavak jelenlétére.



Az árnyékjelszavak általános elterjedésével a kézi beállítások szinte teljesen eltűntek. Ennek ellenére léteznek eszközök, amelyek segítik az árnyékjelszavak kézi beállítását. A `pwconv` és `grpconv` parancsokkal a felhasználókra és a csoportokra vonatkozó bejegyzéseket a /etc/shadow és a /etc/passwd állományokban mindig szinkronban tarthatjuk, de a használatuk általában szükségtelen, mivel ez a művelet az új jelszavak megadásakor önműködően zajlik. Hasonlóképpen a `pwunconv` és `grpunconv` parancsokkal ugyan hagyományos jelszavakat is létrehozhatunk, de a mai korszerű rendszerek esetében ez a visszalépés ritkán szükséges. Talán az egyetlen valóban hasznos eszköz a Debian Linux `shadowconfig` programja, amelynek `off`-on beállításával rendszerünkben gyorsan és könnyen engedélyezhetjük az árnyékjelszavak használatát.
A /etc/shadow állomány többi oszlopa is evolúciós zsákutcának bizonyul. Első pillantásra hasznosnak tűnnek, hiszen

a segítségükkel beállíthatjuk, hogy mennyi idő után kell megváltoztatni egy jelszót, a felhasználó mikor kapjon erre vonatkozó figyelmeztetést, és mennyi idő után tiltsa ki a rendszer, ha mégsem változtatta meg a jelszavát. A fejlődés tekintetében ezek az oszlopok a túlélést elősegítő vonások, azonban ezeket a beállításokat minden egyes felhasználó esetén sajnos egyenként kell elvégeznünk. Még nagyobb hátrány, hogy az időpontokra vonatkozó bejegyzéseket az 1970. január 1-je óta eltelt napok számában kell megadni. Ha pontosak akarunk lenni, ez annyira macerásnak bizonyul, hogy a rendszergazdák többsége az oszlopok nagy részét üresen hagyja, a többibe pedig olyan nagy számokat ír, amelyekkel biztosan nem kell többet foglalkoznia.

Az MD5 kora

Az hagyományos jelszavas védelmi rendszer egy másik továbbfejlesztett utóda az MD5 névre hallgató titkosító eljárás. Az MD5 a legfrissebb termék azon algoritmusok fejlődésében, amelyet ***Ronald Rivest***, az MIT professzora és a titkosító eljárások kifejlesztésében több mint egy évtizede élenjáró cég, az RSA Security alapítója dolgozott ki. Az MD5 a 8-bites gépekre finomhangolt MD2 utóda, és közvetlenül a 32-bites rendszerekre kifejlesztett MD4 algoritmus módosításából jött létre. Rivest és munkatársai szerint az MD4 annak idején túlságosan hamar került a piacra. Az MD5-öst 1991-ben szabadon felhasználható (public domain) eszközként adták ki, majd 1994-ben további módosítására került sor.

Az MD5 algoritmus az azonosítást igénylő védelmi rendszerek szabványává vált, igaz, Rivest kitart amellett, hogy eredetileg nem ezzel a céllal fejlesztették ki. Bár az elmúlt években olyan kifinomultabb titkosító eljárások jelentek meg, mint például az IDEA, a Skipjack vagy a Bowfish, eddig még egyikük sem bizonyította, hogy annyival jobb teljesítménnyel bír, amennyivel az MD5-öt kiüthetné a nyeregből.

A kilencvenes évek közepén az MD5 csak utólagos kiegészítőként kerülhetett a Linux-rendszerekbe, mára azonban a legtöbb terjesztés tartalmazza. Biztonság tekintetében az MD5 előnye a hagyományos jelszavas rendszerekben használt DES-eljárással szemben az, hogy megengedi a hosszabb jelszavak használatát, és kifinomultabb titkosítást tesz lehetővé. Az MD5 engedélyezésével rendszerünk védelmére akár 256 karakter hosszú jelszavakat is használhatunk. A jelszó hosszától függetlenül az MD5 négy lépésben végzi a kódolást, melynek eredménye egy 256 karakterből álló kulcs. Mivel ez a művelet visszafordíthatatlan (legalábbis nem sok rá az esély), az MD5 algoritmust egyirányú hashnek is nevezik. Az ismertebb Linux-változatok telepítéskor felajánlják az MD5 telepítését is.

Bár az MD5 számos mai munkaállomás és hálózat adatrendszereiben gondokat okozhat, ez mégsem ok arra, hogy ne használjuk. Ha nem vagyunk biztosak benne, hogy az MD5 engedélyezve van a rendszerünkben, két módon meggyőződhetünk erről. Nézzük meg, hogy a /etc/shadow állomány jelszóoszlopa a `$1$` karakterekkel kezdődik-e, vagy nézzük bele a ***/etc/pam.d*** könyvtár állományaiba „md5”-re végződő sorok után kutatva. Ha az MD5 nincsen engedélyezve, az ehhez szükséges állományok előkeresése és a rendszer újbóli beállítása általában annyira időigényes folyamat, hogy a legtöbb kezdő felhasználó inkább a teljes rendszer újratelepítése vagy frissítése mellett dönt.

A csontváz tagolása: PAM

Az árnyékjelszavak és az MD5-eljárás elterjedése a bőség zavarát okozhatja, mivel a működésükhöz szükséges utólagos kiegészítők sokféleképpen kombinálhatók egymással, és mindegyikük saját parancsokkal rendelkezik, mint például a `passwd` vagy `login`. Erre a dilemmára nyújt megoldást a Pluggable Authentication Method (körülbelül: „csatlakoztatható hitelesítő eljárás”), röviden a PAM. A PAM tulajdonképpen a védelmi folyamatokban és a biztonsági szintek megváltoztatásában szereplő parancsok és munkafolyamatok közti közvetítő. A PAM-módszer párhuzamosan fejlődött az árnyékjelszavakkal és az MD5-tel, és mintegy 1997 óta része a különböző Linux-változatoknak.

A PAM-ot eredetileg a ***/etc/pam.conf*** állományban lehetett beállítani, a legtöbb terjesztésben ez az állomány azonban már annyira haszontalanná vált, mint testünkben a vakbél. Helyét a /etc/pam.d könyvtár vette át.

A könyvtár egyes állományaiban beállíthatjuk, hogy egy adott parancshoz mely felhasználók vagy csoportok férhetnek hozzá, például a ***/etc/pam.d/su*** állományba a `su` parancsot szabályozó bejegyzéseket tehetünk. A ***/etc/security/limits.conf*** állományban további korlátozásokat állíthatunk be, de a /etc/pam.d legtöbb állománya a ***/lib/security*** függvénykönyvtáraira mutatva közvetítő szerepet lát el a jelszavas biztonsági eszközök és egyéb parancsok között. Ezek közé tartoznak a `chfn`, `chsh`, `cron`, `gdm` és a `login`. Ezáltal nemcsak az árnyékjelszavak és az MD5 használata válik lehetővé, hanem az olyan fejlettebb biztonsági megoldások, mint például a Kerberos bevezetése is leegyszerűsödik.

A /etc/pam.d könyvtárban található állományok összes beállítási lehetőségének részletezése nem áll módunkban, de az állományok bőségesen el vannak látva megjegyzésekkel, a szerkezetük pedig könnyen átlátható. Különös figyelmet érdemelnek a `passwd`, `gdm`, `login` és a `su` állományok, segítségükkel végezhetjük el ugyanis a jelszavas védelem legalapvetőbb beállításait. A `login` állomány szerkesztésével például szabályozhatjuk a rendszergazda bejelentkezéseit, időkorlátot rendelhetünk hozzájuk,

illetve azt is beállíthatjuk, hogy miként történjen a bejelentkezések naplózása. Ha a `su` parancsot használjuk az egyébként több beállítási lehetőséget nyújtó `sudo` helyett, akkor a /etc/pam.d/su állomány lesz a segítségünkre. Bár nem szerencsés megváltoztatni a ***security*** függvénykönyvtárakra való hivatkozásokat, de a rendelkezésünkre álló finomhangolási lehetőségeket érdemes szem előtt tartani – akik számára rendszerük biztonsága kiemelten fontos, valószínűleg nem élnek a `null`-okkal, mivel lehetővé tenné, hogy a felhasználók megváltoztassák az üres jelszavakat. További példaként említhetjük a `chsh` állományt, amelyben a felhasználók által hozzáférhető parancsfájlokat korlátozhatjuk – listájuk a ***/etc/shells*** állományban található. Röviden összefoglalva: előfordulhat, hogy a ***pam.d*** könyvtár rengeteg lehetőségét látva megfájdul a fejünk, az eligazodásra tett erőfeszítésünk azonban mégsem hiábavaló, mert segítséget nyújt, hogy mi módon tehetjük a rendszerünket biztonságosabbá. A PAM témakörét részletesebben is körbejárjuk a 46. oldalon kezdődő cikben.

Fejlődés és új felhasználási területek

Az árnyékjelszavak, az MD5- és PAM-eljárások használata mind biztonságosabbá teheti a rendszerünket. Azt azonban tartsuk figyelmünk gyújtópontjában, hogy a biztonság mindig viszonylagos. Elegendő számítási teljesítőképesség és rendelkezésre álló idő esetén nyílt támadással bármilyen rendszer feltörhető. Ráadásul egyre könnyebbé válik, mivel mind az alkatrészek, mind a betörők által használt eszközök folyamatosan fejlődnek. A jobb átláthatóság kedvéért felidézzük az eseményeket: 1994-ben az RSA Security szakértői úgy becsülték, hogy egy átlagos biztonságú gép elleni nyers számítási erőn alapuló támadás 24 napon belül sikerrel jár. Ezzel szemben az `mdcrack` program fejlesztői (az MD5-eljárással védett rendszerek biztonságát ellenőrző eszköz) állítják, hogy egy 2.2-es változatú Linux-rendszermagot futtató átlagos gépen egy 56 karakterből álló jelszó húsz másodpercen belül feltörhető. Bár ez az idő még mindig kétszer olyan hosszú, mint egy Windows-alapú rendszer feltöréséhez szükséges átlagos időtartam, de már nyilvánvaló, hogy a Linux-felhasználóknak sincs okuk az önelégültségre. A helyzet ráadásul csak romlani fog. A növekvő fenyegetés elleni védekezés egyik legjobb módja, hogy fokozottabban kihasználjuk a jelszavas védelem adta lehetőségeket. Számos, főként otthoni felhasználó a telepítés után rögtön el is feledkezik a jelszavas védelemi rendszerről, és a későbbiekben is csupán a legalapvetőbb szolgáltatásait használja, pedig egy kis odafigyelés a részletekre már elég lenne ahhoz, hogy az őrületbe kergessük a botcsinálta betörőket. Nézzünk néhány gyakorlatból ellesett példát!

- A /etc/shadow állományban adjuk meg, hogy mennyi ideig engedélyezzük egy jelszó használatát. Bár ez elég kényelmetlen megoldás, a rendszeresen váltogatott jelszavakkal mégis visszaverhetjük az olyan támadásokat, amelyek sikerességéhez inkább sok idő kell, semmint számítási teljesítőképesség.
- A /etc/pam.d/passwd állományban mind a legkisebb, mind a legnagyobb jelszóhosszat növeljük meg. Ha minden más feltétel azonos, akkor igaz az állítás, hogy minél hosszabb egy jelszó, annál tovább tart feltörni.
- A /etc/pam.d/gdm állományban csökkentsük a bejelentkezési kísérletek számát. Azok a törvényes felhasználók, akik gyakran vétenek gépelési hibákat, ugyan panaszkodni fognak, de a behatolással próbálkozókat ez valószínűleg annyira megzavarja, hogy inkább odébbállnak.
- Személyesen hagyjuk jóvá a felhasználók által választott jelszavakat, de legalábbis ragaszkodjunk ahhoz, hogy olyan programot használjanak, amely biztonságos jelszavakat hoz létre, mint például a `pwgen`. Elmondani is fájdalmas, milyen sokan vannak, akik jelszavuknak a „jelszó" szót, legkisebbik lányuk vagy aranyhaluk nevét választják.
- Telepítsünk rendszerünkbe olyan programot, mint például a `cracklib2`, ami megakadályozza a könnyen kitalálható jelszavak használatát. Egyénileg összeállíthatunk egy szójegyzéket, amelyben megadhatjuk a nemkívánatos szavak listáját. Ilyenek lehetnek az ismertebb cégek, termékek nevei vagy a gyakori felhasználói nevek. Bár a `cracklib2` helyes beállításához szükségeltetik egy kis türelem, és a /etc/pam.d állomány néhány sora elől is ki kell vennünk a megjegyzéseket, mégis nagyon valószínű, hogy a `cracklib2` vagy a hozzá hasonló programok az elkövetkező években bekerülnek az elterjedtebb Linux-változatokba.
- Ne engedélyezzük, hogy a rendszergazda távolról is bejelentkezhessen.
- Kísérjük figyelemmel, hogy mikor járnak le a jelszavak, és szűrjük ki a jelszó nélküli felhasználókat. Mindkét eset rossz szándékú belépésre ad lehetőséget.
- Kapcsoljuk ki a gépünket vagy állítsuk le a hálózati kapcsolatot, ha már nincs rá szükségünk. Ne játszd a számítógépes nehézfiút, aki folyton azzal kérkedik, hogy milyen hosszú ideje nem kellett újraindítani a rendszerét. Hálózati kapcsolat nélkül a távoli betörések lehetetlenek.

A mai rendszerek kockázatos helyzetére adott válaszként megnövekedett a jelszavas azonosítást igénylő pontok száma rendszeren belül. Ha még nem tettük meg, a következő helyeken alkalmazhatunk jelszavas védelmet:

- BIOS: használjuk a jelszavas védelmet, és győződjünk meg arról, hogy nem kerülhető meg akkor sem, ha a rendszert hajlékonylemezről vagy CD-ről indítjuk.
- Indításkezelők: alkalmazzuk a LILO `password` vagy a GRUB `lock` parancsát.
- Távolról elérhető szolgáltatások: az SSH-protokoll titkosítva küldi el a jelszavakat a hálózaton – a Telnet- és FTP-protokollok viszont nem. Ugye nem kell elmondanunk, hogy melyiket használjuk…

Természetesen csak jelszavas védelemre támaszkodva mindössze a zöldfülű betörők ellen védekezhetünk. Ráadásul minél nehezebb feltörni egy jelszót, annál nehezebb fejben tartani – ez ahhoz vezethet, hogy a felhasználók a bonyolult jelszavaikat egy öntapadós cetlire írják le, amit azután felragasztanak a billentyűzetük aljára.
Még sincs semmi okunk, hogy ne használjuk a jelszavas védelem kínálta biztonsági előnyöket. Azt pedig végképp kerüljük el, hogy kiiktassuk vagy gyengítsük a jelszavas védelmet – sajnos ez a jelenség kezdi felütni a fejét a mai Linux-változatokban abból a célból, hogy a Linux jobban hasonlítson az átlagfelhasználó által használt operációs rendszerekhez. Az eszközök a kezünkben vannak, rajtunk áll, hogy használjuk-e őket!

Bruce Byfield (bbyfield@axionet.com) szabadúszó szakíró, termékmenedzser, újságíró. Amikor nem a számítógépe előtt ül, egzotikus madarak társaságában mozog, punk-folk zenét hallgat, és kedvtelésből fájdalmasan hosszú távokat fut.