Könnyű álmok (2. rész)

Nézzük végig az alapvető biztonsági kérdéseket
a hálózattervezés, az operációs rendszer telepítése és a programfejlesztések háza táján.

Mielőtt rátérnénk a hálózati biztonsági veszélyforrások ismertetésére,meg kell értetnünk ezek kiváltóit. Az okok feltárása nélkül nehezen érthető meg az, hogy mi ellen kell megvédenünk rendszerünket. A leggyakrabban olyan – a programfejlesztők által elkövetett – tervezési vagy kivitelezési hibák okozzák a rendszerek támadhatóságát, amelyek a korszerű tervezési módszerek alkalmazásával és némi ráfordítással elkerülhetők. Miért kell mégis félnünk a betörésektől? Mert a fejlesztő cégek a költségek csökkentéséért nem alkalmazzák ezen módszereket. A hálózati rendszerek tervezői és kivitelezői pedig hajlamosak a tapasztalatukra, vélt szaktudásukra hagyatkozni. Ha az ember figyeli a biztonsággal foglalkozó levelezési listákat (*lásd a táblázatban*), akkor nap mint nap tapasztalhatja, hogy a nagy és drága rendszerek tervezői is fittyet hánynak ezen elméletek betartására, és olyan alapvető biztonsági hibákat hagynak rendszerükben, amit a hasonló célú rendszerekből már évekkel ezelőtt kifogtak. Ha a rendszerek fejlesztői csak arra vennék a fáradságot, hogy a korábbi fejlesztési tapasztalatokat megismerjék, a biztonsági hibák nem lennének olyanok, mint a visszajáró kísértetek. Jó példa az a hiba, amit a webkiszolgálók fejlesztői szinte kivétel nélkül meghagynak a rendszerükben: a szakma csak úgy hívja, a pont-pont hiba (dot-dot bug). Ennek az a lényege, hogy a webkiszolgáló nem ellenőrzi az elérési útban található "../" karaktereket, vagy azok kódolt változatát, így nemcsak a beállítófájlban engedélyezett könyvtárak tartalmához lehet hozzáférni, hanem bármihez. Ez a jellegzetes „állatorvosi ló" esete. Havonta van olyan HTTP protokollt használó eszköz, melyben felfedezik ezt a hibát. Olyan webkiszolgáló is létezik, ahol egyszer kijavították, aztán visszakerült.

A nyílt forráskód előnyei

Amikor egy hiba napvilágra kerül, a rendszer fejlesztőjének joga van eldönteni, hogy mikor javítja ki. Ezzel korábban igen komoly nehézségek akadtak, mivel a felderített hibákat a fejlesztők lassan vagy egyáltalán nem javították ki. A biztonsággal foglalkozó levelezési listák nagy előnye, hogy sikerült kikényszeríteni a nagy programgyártó cégekből is a gyors visszahatást, hiszen ha itt megjelenik egy hiba, akkor azt azonnal ki kell javítani. Nagyságrendekkel jobb a helyzet a nyílt forráskódú rendszereknél, ugyanis mivel a kód nyílt, a fejlesztők nem helyezhetnek el semmilyen meglepetést (olyan programrészt, amely például adatokat továbbít a fejlesztőcégnek). A nyílt forráskód következtében egyszerűbb egy átfogó kódvizsgálat alá vetni, ezeket a rendszereket (léteznek ilyen kezdeményezések). A nyitott forráskódú rendszerek sem mentesek a hibáktól, azonban a forráskód nyíltságának köszönhetően a hibák javítása itt lényegesebben gyorsabb. Ha egy levelezési listán egy rendszer hibájáról olvashatunk, a hiba felfedője gyakran elküldi a javítást is.
Nézzük most az általánosan jelentkező hibákat.

Rendszertervezési hibák

Ma már elfogadott tény, hogy komolyabb rendszert megfelelő minőségben nem lehet létrehozni átfogó tervezés híján. A rendszer elfogadható minőségéhez pedig ugyanúgy hozzátartozik a biztonsági szempontok figyelembevétele, mint a kis erőforrásigény vagy a hibátlan adattárolás. A tervezési munkát megkönnyítendő a számítástechnikában is létrejöttek programtervezési és fejlesztési módszertanok. Ezek leírják, hogy a programtervezőknek és fejlesztőknek milyen lépéseket kell megtenniük ahhoz, hogy a végtermék a várt minőségű legyen. Akkor járunk a legközelebb az igazsághoz, ha úgy képzeljük el ezeket a módszertanokat, mint az informatika minőségbiztosítási szabványrendszerét – ugyan itt nincs elfogadott szabvány. Ha egy rendszert valamelyik módszertan szerint terveztek és fejlesztettek, akkor biztosak lehetünk benne, hogy abban nincs több hiba, mint amennyit az adott módszertan megenged. Több ilyen módszertan létezik, minden rendszer jellemzője azonban, hogy a szolgáltatásgazdagságra összpontosít (azon belül is elsősorban adatközpontú), és általában nem szentel elegendő figyelmet a biztonságnak.

Fejlesztési hibák

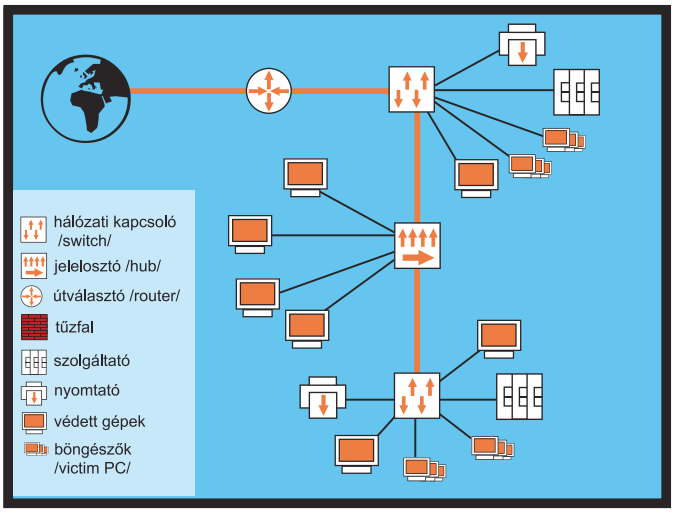
Amennyiben a rendszer terve megfelelő, a labda a rendszerfejlesztőké. Ha ők valamilyen hibát ejtenek, akkor a rendszer a tervezők lelkiismeretes munkája ellenére is sebezhető lesz. A legtöbb programozási hiba sokszor ismétlődik, így a fejlesztők fel tudnak – tudnának – készülni ellene. Ilyen elkövetett programozási hibák minden programozási nyelvben megtalálhatók. Vannak ugyan olyan leírások is, melyek a hiba elkerülésének útját írják le [1., 2.], ezek elolvasására azonban nem minden fejlesztéssel foglalkozó cég kötelezi a fejlesztőit. Ha tanulmányoznák ezeket az anyagokat, akkor a hibák jó része kiküszöbölhető lenne. A forráskód felülvizsgálatakor ezek a hibák megfoghatók, azonban ennek lényegesen komolyabb erőforrásigénye van, így célszerűbb már a fejlesztés idején nem hibázni.

Common Criteria

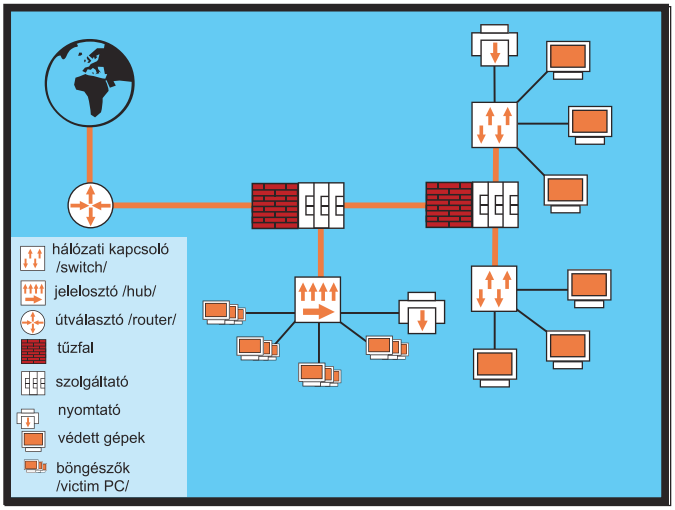
Ha egy program fejlesztésénél a biztonság fontos szempont, akkor célszerű a CC (Common Criteria – Általános elvárások) [3.] időszerű változatát figyelembe venni a tervezésnél és fejlesztésnél. A CC egy nemzetközileg elfogadott fogalom- és követelményrendszer, amelyet korábban az egyes országok saját követelményrendszereként használt belső szabványokból (többek között a TCSEC-ből) fejlesztettek tovább. Jelenleg még nem minden országban elfogadott, de a jelentősebb informatikabiztonsági fejlesztők már figyelembe veszik, esetenként a fejlesztés teljes egészében a CC iránymutatásai szerint folyik. A felhasználói számára biztosítja, hogy az egyes biztonságtechnikai fogalmak minden tervező és fejlesztő számára ugyanazt jelentsék, és a termékek biztonsága nemzetközileg is mérhető legyen. Ha egy rendszer fejlesztői el akarják ismertetni terméküket mint CC alapján bevizsgált eszközt, akkor rákényszerülnek, hogy a rendszerre leselkedő veszélyeket felderítsék és elhárítsák.

Követelmények

A CC hét garanciaszintet határoz meg, ezek egyre erősebb intézkedésekkel kényszerítik a fejlesztőket arra, hogy az eredeti elképzelések szerint, hibátlanul működő rendszert hozzanak létre. Tehát a garanciaszint *nem befolyásolja* a biztonsági szintet, csupán biztosítja, hogy a program csak a tervezői által elképzelt szolgáltatásokat nyújtja. Az alacsonyabb szintek követelményei viszonylag egyszerűen teljesíthetők (például kötelező jó minőségű felhasználói leírást szállítani a rendszerrel), a harmadik és negyedik szint már erősebb követelmé-



1. ábra Rosszul tervezett hálózat vázlata



2. ábra Jól tervezett hálózat vázlata

nyeket támaszt (többek között átfogó ellenőrzést kell végezni), míg a hetedik szinten olyan követelményeknek kell megfelelni, melyek kielégítéséhez igen jó szakemberekre és komoly energiabefektetésre van szükség. Követelmény például az, hogy minden algoritmus helyességét matematikai módszerekkel bizonyítani kell(!).
Belátható, hogy a magasabb garanciaszintek lényegesen magasabb szintű biztonságot adnak ugyan (jó tervezés mellett), de sokkal nagyobb az erőforrásigényük. Nagyobb mennyiségű és általában jobb minőségű munkaerőre van szükség – így nő a költség is. Mivel azonban a költség növekszik, és ma még szinte minden cég elsődleges célja a lehető legnagyobb nyereség elérése, így kevés a CC szerint fejlesztett rendszer. Ha egy rendszer minősítetten megfelel a CC valamely szintjének, akkor lényegesen nyugodtabbak lehetnek a rendszer felhasználói: ennél a fejlesztésnél a biztonsági szempontokra is gondosan ügyeltek.
Különösen kellemetlen hibák az operációs rendszerek tervezési hibái (ideértve a különböző Linux-változatok hibáit is).

Az operációs rendszer

Minden számítógép működésének alapja az operációs rendszer, ennek szempontunkból legfontosabb jellemzői: többfelhasználós, több feladat párhuzamos végrehajtására képes, és számítógépes hálózatot tud kezelni. Amennyiben ezek a tulajdonságok jelen vannak – ez ma már szinte minden operációs rendszerre igaz –, akkor a rendszer veszélyeztetett. Ha tervezési vagy telepítési hibák miatt biztonsági hiányosságok adódnak, akkor a rendszerbe betolakodók hatolhatnak be. Ha a rendszer nem megfelelően felügyelt, vagy erre nincs lehetőség, akkor a kalóz azt tesz az ott található adatokkal és erőforrásokkal, amit akar. Ha a behatolók elég felkészültek, az üzemeltetők nem értenek a dolgukhoz a szükséges szinten vagy maga a rendszer nem ad lehetőséget a teljes felügyeletre, a rendszer jogos felhasználói a behatolást észre sem veszik. Erre az esetre fejlesztik a behatolásérzékelő rendszereket (IDS – Intrusion Detection System [4.]), mézes bödönöket (honey pot) és más eszközöket, de ez egy másik mese lesz. Ha a rendszerben felfednek egy rést, akkor általában csak az operációs rendszer vagy az adott részek frissítése segíthet. Bizonyos rendszereken ezt Szervizcsomagként, máshol „patchmatrix”-ként emlegetik, a Linux-rendszerben – erős modularitása következtében – elegendő a megfelelő csomagok frissítéséről gondoskodni.

1. ábra
Néhány a felmerülő gondok közül: a rendszerben középen elhelyezkedő jelelosztóhoz csatlakoztatott ügyfelek lehallgathatják a két hálózatrész forgalmát, sőt, akár meg is támadhatják valamelyik rendszert, miközben a másik rendszernek adják ki magukat (spoofing). A két hálózati kapcsoló között kifeszített virtuális hálózatok (VLAN) is támadhatók.

2. ábra
A rendszer logikai részei fizikailag is jól elkülönítettek. A két szolgáltató tűzfalakra jellemző feladatokat is ellát, így a rendszerek elérhetősége finoman szabályozható. A világhálóval érintkező gépen a kívülről is elérhető szolgáltatások száma minimalizálható. A megvalósításhoz elég egyetlen nyilvános IP-cím.

Az operációs rendszer hibái

Az operációs rendszerben öt szinten fordulhat elő hiba: a rendszermagban, függvénykönyvtáraiban (lib vagy dll), az alaprendszer futtatható segédprogramjaiban, a rendszerdémonokban vagy a rendszer alapbeállításaiban. Szerencsére elmondható, hogy a Linux rendszermag stabil változatában nagyon ritkán találnak komoly biztonsági rést. A legutóbbi hibát 2000. június 7-én hozták nyilvánosságra. A hiba a 2.2.15-ös rendszermagban volt és a 2.2.16pre6 számúban azonnal javították, illetve elérhető volt egy olyan rendszermodul, amely segítségével a 2.2.15-ös rendszereken sem volt a hiba kihasználható. A lehetőség megragadásával egy rendszer helyi felhasználója bizonyos körülmények között rendszergazdai jogokra tehetett szert. A rendszer függvénykönyvtárairól is viszonylag ritkán derül ki hiányosság (sajnos, az utóbbi időben volt rá néhány példa), a rendszer alapvető segédprogramjai ritkán tartalmaznak biztonsági réseket, a Linux-változatokat pedig nagy gonddal tervezik, így a telepítésnél ritka az olyan biztonsági rés, amely az operációs rendszer alapbeállításainak hibájából adódik. Más rendszerek gyakran igen komoly biztonsági réseket tartalmaznak, melyek folyamatosan kerülnek felszínre, így e rendszerek felhasználói soha nem alhatnak nyugodtan. A helyzetet súlyosbítja, hogy az említett rendszerek forráskódja általában nem érhető el, így nem lehetséges egy átfogó forráskódvizsgálattal a jelentősebb hibák kiszűrése. A zárt forráskód miatt a hibák javítása is nehézkesebb. Sajnos, még mindig kerülnek használatba olyan operációs rendszerek, melyeken adott felhasználóknak – biztonsági vagy bemutató célzattal – előre ismert jelszavuk van, ezzel azonban megkönnyítik a kalózok dolgát.

Telepítés – a karácsonyfa modell

Biztonsági szempontból kényes pont a telepítés. Ha az operációs rendszer telepítőkészlete lehetővé teszi a könnyű telepítést – rengeteg szolgáltatással, akkor gyakorlatlan felhasználók kevés tapasztalattal is nekiállnak és telepítenek. Így születnek azok a rendszerek, amit Lilo barátunk így jellemzett: – „Karácsonykor odamegyünk és letesszük

alá az ajándékokat”. Egyszóval karácsonyfa. Ezzel a kettősséggel küzdenek a Linux változatai: ha a rendszert népszerűbbé akarják tenni, akkor a felhasználók dolgát a telepítésnél meg kell könnyíteniük, ha biztonságra törekszenek, akkor a rendszer telepítésekor a használhatóságot a lehető legbiztonságosabban kell elérni. Régi mondás: „Bármely rendszer biztonsága fokozható a teljes használhatatlanságig.” Ez aztán ellentmondás a javából. Valamit segítenek azok a telepítőkészletek, melyek kézen fogva vezetik a felhasználót a szükségesnek vélt szolgáltatások telepítéséig, azonban a tapasztalatlan felhasználó hajlamos olyan eszközöket is feltelepíteni, melyre nincs szüksége. Azt gondolják: – „Jól hangzik a neve, majd kipróbálom, mire jó!”. Ezeknek a programoknak a nagy részét soha nem indítják el.

Csomagok és kezelők

Az egyedi telepítés kiválasztásával talán elkerülhető lenne a jól ismert „Felhasználói munkaállomás” típustelepítés, ez azonban hosszadalmas. A típusos telepítés viszont magával hozza azokat a típushibákat is, amelyeket a terjesztés figyelmetlen összeállítói esetleg benne hagytak a telepítőkészletben. A csomagkezelőből adódó nézőpontbeli különbség a két legnépszerűbb Linux-változat között: a Debian rendszere (deb kiterjesztésű csomagok) az adott alrendszer telepítésekor megkísérli annak beállítását és elindítását is. Ennek a megoldásnak sajnálatos mellékhatása, hogy egy csomag telepítésekor olyan szolgáltatások is elindulhatnak, amelyeket a rendszer gazdája egyelőre nem akart elindítani. Ha egyszer sikeresen leállítottunk egy ilyen szolgáltatást és arról is gondoskodtunk, hogy a rendszer indulásakor se induljon újra automatikusan, akkor nincs is több gond vele – a következő frissítésig. Akkor ugyanis újraindulnak a szolgáltatások. Ez a dpkg csomagkezelő program jelen változatában csak komoly kommandózással küszöbölhető ki, mivel az kizárólag a csomagban tárolt utasításokat hajlandó elfogadni.

A Red Hat által fejlesztett csomagkezelő rendszer (rpm kiterjesztésű csomagok) a felhasználók szempontjából kissé kényelmetlenebb. A csomagok telepítésekor nem kényszeríti ki azok beállítását, és a rendszergazdára bízza annak eldöntését, hogy az adott szolgáltatást el akarja-e indítani. Biztonsági szempontból mindenképpen az utóbbi eljárás a hatékonyabb. A Debian csomagokban lévő beállítási állományok alapértékei általában biztonságosabbnak mondhatók, és ki tudna lemondani az apt-get parancs kényelméről. Ezek után mindenki döntse el, mit választ. Biztonsági szempontból mindkettő megfelelően használható, ha valaki tapasztalt, akkor mindkét rendszert biztonságossá tudja tenni. Ha valaki önállóan kezd egy Linux-rendszer telepítéséhez, és szempont a biztonság (ha valamilyen módon hálózathoz fog csatlakozni) akkor javasoljuk csak azokat az alrendszereket telepíteni, amelyek valóban szükségesek. Mindenki döntse el, hogy pontosan mire szeretné használni a rendszerét, és ennek függvényében válogasson a rendelkezésre álló csomagok közül. Kezdetben célszerű csak a szükséges és ismert csomagokat telepíteni, később szükség esetén bővíteni lehet az elérhető rendszerek számát. A csomagkezelők függőségkezelő rendszere segít, mivel telepíti azokat a csomagokat, amelyekre a kiválasztott alrendszer működéséhez szükség van.

Biztonsággal foglalkozó levelezőlisták

LEVELEZŐLISTA CÍME	HOL LEHET FELIRATKOZNI
bugtraq@securityfocus.com	http://www.securityfocus.com
vuln-dev@lists.securityfocus.com	http://www.securityfocus.com
secprog@securityfocus.com	http://www.securityfocus.com
sf-news@securityfocus.com	http://www.securityfocus.com
firewall-wizards@nfr.com	http://www.nfr.com/forum/firewall-wizards.html
debian-security@debian.org	http://www.debian.org/MailingLists/subscribe
debian-security-announce@lists.debian.org	http://www.debian.org/MailingLists/subscribe
linux-security@redhat.com	https://listman.redhat.com/mailman/listinfo/linux-security
suse-security@suse.com	http://www.suse.com/us/support/mailinglists/index.html
security-l@sunserv.kfki.hu	http://sunserv.kfki.hu/mailman/listinfo/security-l

A hálózati felépítés hibái

Ha a számítógépes hálózat tervét nem ellenőrzi biztonsági területen jártas szakember, akkor bizony előfordulhat az alábbi elképzelt (sajnos nem ritka) eset: egy cég éppen új irodába költözik. A vezető a költségek csökkentése miatt olyan céget bíz meg a hálózat tervezésével és kivitelezésével, amelynek hihető referenciái ugyan nincsenek, de saját állítása szerint jó a szakmában és – a cégnek elsősorban ez számít – olcsón dolgozik. A cég hálózatát a lehető legegyszerűbb felépítésűre tervezik. (*1. ábra*) A tervek szerint a könyveléssel foglalkozó részleget nem lehetséges leválasztani a cég programfejlesztő részlegétől. Ha a cég valamelyik fejlesztőmérnöke rosszindulatú, hozzáférhet a fizetési adatokhoz, akár módosíthatja is azokat. Ha a hálózat helyesen lenne tervezve (*2. ábra*), akkor a két hálózat fizikailag leválasztható lenne egymásról, és a pénzügyi részleg egy tűzfallal védhető lenne. Ha egy hálózaton van minden gép, akkor megvalósítható a cikksorozat előző részében ismertetett lehallgatás. Ennek elkerülésére szokták – szintén költségmegtakarítási okból – a hálózatot szétválasztani virtuális hálózatokra (VLAN). Ez azt jelenti, hogy a hálózatban lévő aktív eszközökkel (leggyakrabban hálózati kapcsolókkal) a forgalmat úgy irányítják, hogy az egyes virtuális hálózatokban lévő gépek csak a velük azonos hálózatban lévő gépeket láthassák. Van azonban egy kis baj: amennyiben a kapcsoló vezérlőprogramjában biztonsági rés van (sajnos, többször is előfordult már), akkor az egyes virtuális hálózaton lévő gépek a hálózati kapcsoló megtévesztésével láthatják más VLAN-on elhelyezkedő gépek forgalmát, rossz esetben módosíthatják is azt. Miután a rosszindulat a hálózatot újra egyesítette, a lehallgatás már kivitelezhető, vagy megvalósítható egy középre belépéses („man-in-the-middle”) támadás is. Jelenleg általánosan elfogadott álláspont: a hálózati kapcsolók (switch) nem biztonsági eszközök. Tervezésükkor általában nem a biztonság az elsődleges szempont.

Hibás rendszerek használata

Biztonsági hibákat lépten-nyomon találnak szinte minden rendszerben. Unatkozó varázslók (hacker, geek) találják meg, vagy kalózok, ez csak szerencse kérdése. Ha varázslók, akkor a hibát először a fejlesztőkkel közlik, általában a javítással együtt, és várják a hiba és a megoldás közzétételét. Ezek után – jó esetben – a fejlesztők értesítik a nyilvánosságot, hogy a rendszerben olyan biztonsági gond merült fel, amely veszélyezteti a használók biztonságát. Amennyiben a fejlesztők nem válaszolnak, akkor a hiba felfedezői elküldik annak részletes leírását, a bemutatására alkalmas programot és a javítókódot valamelyik biztonsági levelezőlistára. Amikor a hiba napvilágot lát, akkor adódhat egy kis bökkenő. Ha ugyanis egy felhasználó nem olvassa az adott listát, akkor rendszere átmenetileg védtelen a behatolók ellen. Az ilyen felfedezett, de az adott rendszeren még nem kijavított hibák a betörések leggyakoribb okai. Ha a hibát nem is egyszerű kihasználni, a szemléltető program felhasználásával, vagy annak kis módosításával betörőeszköz nyerhető. Ha a betörő nem rendelkezik elegendő szakértelemmel, akkor a hiba tudatában is tehetetlen lenne, a jóindulatú bemutatóprogram segítségével azonban... Az ilyen kisebb szakértelemmel, mások által fejlesztett eszközök segítségével behatoló kalózokat hívja az angol szaknyelv „script kiddie”-nek.

Ennek az általánosan elterjedt gyakorlatnak a megállítására egyre gyakrabban csak a szemléltető program vázát adja közre a hiba felderítője, így a hozzáértők látják a gondot, a fent említett hozzá nem értők viszont segítség nélkül nem tudnak mit kezdeni a példa-

programmal. A hibák ilyen szemléltetése arra mindenképpen jó, hogy szükség esetén rákényszerítse a rendszerek fejlesztőit a hiba mielőbbi javítására. Ha a hibára kalózok akadnak rá, akkor a helyzet lényegesen súlyosabb. Míg az adott résen való behatolást nem sikerül valakinek észlelnie, addig zavartalanul járkálhatnak ki-be azokon a rendszereken, melyek a hibás alrendszert tartalmazzák. Jó példa az akkortájt egyik legismertebb biztonsággal foglalkozó nemzetközi weblap ➲ www.rootshell.com feltörése, ahol a betörés módjára csak hónapokkal később sikerült rájönni. (Az incidens rövid ismertetése a [11.] *hivatkozásnál* található.) A gondot súlyosbítja, hogy a kalózoknak kiterjedt nemzetközi betörőprogram-cserehálózata van.
Abban az esetben, ha a rendszer forráskódja nyílt, akkor lehetőség nyílik olyan megelőző intézkedések végrehajtására, melyek a hibák felkutatására irányulnak. Jelenleg is folynak ilyen irányú megelőző erőfeszítések az LSAP (Linux Security Audit Project) [5.], az OpenBSD Security [6.] és számos más kezdeményezés keretében.

Beállítási hibák

Ha a rendszert sikerült megfelelően megtervezni, kifejleszteni és a telepítés is jó, akkor már csak a beállításoknál véthetünk hibát. Nagyon gyakori hiba a következő: egy közepes cég eljut arra a szintre, hogy már feleslegesen sok az ISDN vonal használatából adódó havi költségük. Úgy döntenek, hogy bérelt vonali kapcsolatot építtetnek egy Linux-alapú tűzfallal, amin keresztül leveleznek és böngésznek. A megbízott cég olcsón dolgozik és látszik rajtuk, hogy nagyon értenek hozzá, mert olyan szavakat használnak beszéd közben, mint „lokálinterfész” meg „ípécsénszrúl” és így tovább. Megteszik. Elmennek. Minden ment, mint a karikacsapás. Egy nap csörög a telefon, és egy kedves üzlettárs közli, hogy nem kapja meg a levelet cégünk vezetőjétől, amit pedig régen elküldött, és az ő rendszergazdájuk azt üzente, hogy állítsák be rendesen a levelezőkiszolgálót, mert felkerültek valami feketelistára... Elég homályos ügy. Mi történt? Hiszen eddig működött...
A hiba oka az volt, hogy a jó minőségű levelezőrendszert nem elég hozzáértéssel állították be, és a rendszer elfogadott olyan levelet is továbbításra (spam), amely nem neki szólt. Ezt hívják nyitott átjárónak (angolul: open relay), és lehetővé teszi levélszemét továbbítását anélkül, hogy a küldőnek saját rendszerét le kellene terhelnie. Eljuttatja a levél tartalmát és a címlistát a rosszul beállított rendszerre, az pedig szorgalmasan kiszórja a megadott címekre. Ha ilyen rendszertől kapunk levelet, akkor az jó eséllyel érdektelen hirdetési vagy propagandaanyag, így több olyan szervezet is létrejött, mely célja ezen rendszerek működésének akadályozása. Az alapötlet zseniális: az ellenőrző rendszer képes arra, hogy megállapítsa egy szolgáltatóról a nyitottság tényét, és ha egy rendszert nyitottnak talál, akkor azt feljegyzi. Ha valaki használni kívánja ezt a szolgáltatást, akkor időnként letölti az éppen időszerű listát, és a listán szereplő kiszolgálóktól nem fogad el levelet. Mikor a hálózaton valaki észleli egy rendszerről, hogy nyitva áll az idegen levelek előtt, akkor jobb esetben szól a rendszer postamesterének. Ha azonban azonnal az ilyen rendszereket nyilvántartó ORBS (Open Relay Behaviour-modification System) [7.] vagy MAPS (Mail Abuse Prevention System LLC) vagy RBL (Realtime Blackhole List) [8] rendszereket értesíti, akkor az ezeket használó rendszerek nem fogadnak el tőle levelet mindaddig, míg a beállítás ki nincs javítva.
Az eset tanulsága, hogy egy jól tervezett és megvalósított szolgáltatórendszert is be lehet állítani úgy, hogy a rendszer biztonsága durván sérül. Sok példát lehetne még hozni, de nincs értelme, hisz mindenki látja, mi a hiba. Egy rendszer telepítése előtt annak leírását nagyon alaposan át kell tanulmányozni, különben a beállításokban komoly hibát véthetünk. A rendszerek általában tartalmaznak valamilyen beállítórendszert is, ebben azonban az esetek nagy többségében nem érdemes megbízni. Érdemes minden beállítást leellenőrizni, hiszen ezeknek az automatáknak a tervezői nem gondolhattak minden lehetőségre és általában a rendszer későbbi biztonsági bővítései sem állíthatók be velük. Egy ilyen automata beállító rendszer biztonsági szempontból csak a rendszer működőképességét mutatja be, a finomhangolást kézzel kell elvégezni.

Összegzés

A rendszer biztonsága tehát a tervező–fejlesztő–felhasználó szentháromság jó összmunkáján alapszik. Ha valamelyik láncszem nem elég erős, akkor a rendszer sebezhető lesz. Ha szolgáltató vagy védelmi rendszert kell választanunk, akkor célszerű meggyőződni arról, hogy egy *független* szakértőkből álló csapat mit mond az adott rendszerről. A fejlesztők gyakran elfogultak a saját termékükkel szemben. Ha a rendszert egy országosan, esetleg nemzetközileg elismert csoport biztonságosnak ítéli, akkor jó esély van rá, hogy az alapvető biztonsági követelményeknek megfelel. *De nem tökéletes.* Hiteles szakember nem állíthatja, hogy az általa fejlesztett rendszerben nincs hiba. Csak akkor lehet egy rendszer biztonsága tökéletes, ha az nem csinál semmit. Ennek az elvnek kitűnő bemutatása a tökéletes tűzfal leírása, melyet a szakma egyik legnagyobb embere, Marcus J. Ranum készített [10.]. Ami meglepő: a tökéletes tűzfal nem drága és a leírás alapján akár egy avatatlan is képes a telepítésére. Mindössze egyetlen gond van vele: szerszámboltban árulják és a magyar neve csípőfogó.
Folytatjuk.

Hivatkozások:
[1.] Secure UNIX Programming FAQ: ➲ http://www.whitefang.com/sup/
[2.] WWW Security FAQ: ➲ http://www.w3.org/Security/Faq/www-security-faq.html
[3.] Common Criteria: ➲ http://www.commoncriteria.org/
[4.] Linux Intrusion Detection System: ➲ http://www.lids.org/
[5.] Linux Security Audit Project: ➲ http://lsap.org
[6.] OpenBSD security audit: ➲ http://www.OpenBSD.org
[7.] ORBS (Open Relay Behaviour-modification System): ➲ http://www.orbs.org/
[8.] MAPS (Mail Abuse Prevention System LLC) RBL (Realtime Blackhole List): ➲ http://mail-abuse.org/rbl/
[9.] Marcus J. Ranum: http://web.ranum.com/
[10.] Marcus J. Ranum: The ULTIMATELY Secure Firewall: ➲ http://web.ranum.com/pubs/a1fwall/index.htm
[11.] ➲ http://www.linuxvilag.hu/cikkek/2000dec/konnyu2/

Mátó Péter (atya@andrews.hu), informatikus mérnök és tanár. Biztonsági rendszerek ellenőrzésével és telepítésével, valamint oktatással foglalkozik. 1995-ben találkozott először linuxos rendszerrel. Ha teheti, kirándul vagy olvas.

Borbély Zoltán (bozo@andrews.hu), okleveles mérnök-informatikus. Főként Linuxon futó számítógépes biztonsági rendszerek tervezésével és fejlesztésével foglalkozik.
A 1.0.9-es rendszermag ideje óta linuxozik. Szabadidejét barátaival tölti.

A főszerkesztő ezúton kér elnézést a tisztelt olvasótól és a szerzőktől, ha úgy érzik, hogy a szerzők „technicus terminusainak” magyarítása csorbította a szöveg érthetőségét.