IPCop – Egy hálózsaru feljegyzései

Tilthatunk és engedélyezhetünk – például adott honlapok látogatását, s kis odafigyeléssel megelőzhetjük a betörési és fertőzési kísérleteket, ha feltelepítjük a megfelelő kiegészítő modulokat, vagy kézzel megírjuk a megfelelő szabályokat.

© Kiskapu Kft. Minden jog fenntartva

Kezdeti örömünket (működik a tűzfalszoftver!) gyorsan elronthatja az aggodalom: meddig marad ez védhető és jól prosperáló rendszer? A jó hír, hogy ez nagymértékben rajtunk múlik, a rossz pedig, hogy nem csak rajtunk. Mind a hálózati cím, mind a tevékenységek „ujjlenyomata" hamisítható, csakúgy, mint a felhasználók kiléte. Nem kell elkeseredni, inkább próbáljuk meg átgondolni, mi a teendőnk ahhoz, hogy optimálisan működjön a hálózat, s minél kisebb támadási felületet hagyjunk a rosszindulatú próbálkozóknak.
A következőkben néhány háttérinformációval, működésbeli sajátossággal foglalkozunk. Ezek módosítgatása első ránézésre kevésbé tűnik könnyed és egyszerű mozdulatnak, mint a grafikus felületen beállítani a programok viselkedését. Ne csüggedjünk, majdnem mindnek van kevésbé kifinomult, de könnyebben használható grafikus alternatívája. Ha kicsit megismerkedünk a parancssori változatokkal, látni fogjuk, mennyi terhet levesz vállunkról a könnyen beállítható ***IPCop*** és a hozzá írott sok beépülő modul.

Támadás kívülről

A tűzfalon kívüli veszélyforrások jelentik sokak szemében a legnyilvánvalóbb fenyegetést, noha egy belső hálózaton garázdálkodó egyén sokszor nagyságrendekkel nagyobb károkat képes okozni. Éppen ezért a támadások kivédésére elsődlegesen egy megfelelő biztonsági szabályzatot érdemes megfogalmazni, majd azt következetesen alkalmazni. Ezek után már elkezdhetünk azon is aggódni, vajon nem akar-e valaki jogosulatlanul, netán rosszindulatúan hozzáférni erőforrásainkhoz?
A tűzfal általában a legelső védelmi vonalat képviseli belső hálózatunk és az internet „vad világa" között. Normális esetben a tűzfal – akár szoftveres, akár hardveres megoldás – tartalmának értéke nem fogható a belső hálózaton található szerverek és kliensek információs bázisaihoz, tehát nem magát a tűzfalgépet kell félteni, hanem a védendő objektumokat. Azt azonban figyelemmel kell kísérni, hogy nem vált-e korrupttá a tűzfal (betörés, fertőzés, hardver- és szoftverhiba stb.), és történt-e betörési vagy kitörési kísérlet, s mivel esetünkben az ***IPCop*** útvonalválasztó (***router***) is egyben, korántsem mindegy, hogy valaki hozzányúlt-e a beállításaihoz. Előfordulhatna például, hogy egy rosszindulatú betolakodó hozzáfért rendszerünkhöz, s az útvonalválasztó tábla manipulálásával a kimenő és bejövő hálózati csomagokat az általa felügyelt számítógépre irányítva lehallgatja kommunikációt.
Ez ellen szerencsére a rendszer többszintű védelmet biztosít, s magunk is ellenőrizhetjük a beállításokat.

Csomagszűrés tűzfal módra

Nagyon leegyszerűsítve a ***TCP/IP*** protokoll szerint zajló – például internetes – kommunikációról azt mondhatjuk, hogy az alkalmazások kiválasztanak egy számmal azonosítható nyitott kaput a rendszeren – ezek az ún. portok – s ezen keresztül áramlanak az adatok. Ha ezt a kaput bezárjuk, az alkalmazás vagy nem képes ki- s bejuttatni a hálózati csomagokat, vagy kénytelen magának más portot választani – ha felkészítették erre az eshetőségre. 0 és 1024 között találhatók az úgynevezett jól ismert portok, ezek és egyéb tipikusnak mondható azonosítókat s a hozzájuk tartozó protokoll megnevezését megtalálhatjuk a ➲ ***www.iana.org/assignments/port-numbers*** címen.
Kernelszintű védelemmel rendelkeznek a Linux disztribúciók, a 2.4.x-es verziószám utáni változataikban már bizonyosan az ***Iptables*** elnevezésű változattal találkozhatunk; ezek a hálózati forgalmat „láncokba" fűzve, azaz szabályrendszerek szerint terelik. Három ilyen alapvető lánc a `FORWARD`, `INPUT` és `OUTPUT`, de ha kiadjuk az

```
iptables -L
```

utasítást, láthatjuk, hogy az ***IPCop*** ennél többet is használ.
A láncokon folyamatosan keresztülhaladva a megadott szabályok szerint történik annak elbírálása, mit kell tenni az adott fejlécet tartalmazó hálózati csomaggal. Ha nincs illeszkedő szabályrendszer, akkor az adott lánc alapértelmezett beállítása szerint engedélyezi (`ACCEPT`) vagy hibaüzenettel eldobja (`REJECT`) vagy csak csendben eldobja (`DROP`). Mindezt naplózással (`LOG`) illetve egyéb irányítással (`SNAT`, `DNAT`, `MASQUERADE`) is kezelhetjük, ráadásul a láncokat táblákká összefogva kezeli a rendszer, de ne bonyolódjunk jobban bele, nézzük csak az alapértelmezett csomagszűrést, a ***Filter*** táblát.
Ha a semmiből kellene tűzfalat készítenünk egy ***Linuxot*** futtató számítógép segítségével, akkor jó esetben úgy vágnánk neki, hogy minden letiltunk, majd a legszükségesebb szolgáltatáso-

kat engedélyezzük. Ez az `iptables` program segítségével nagyjából a következőképpen menne. Alapértelmezett irányelvként megadtuk a csomagok csendes eldobását:

```
iptables -P FORWARD DROP
iptables -P INPUT DROP
iptables -P OUTPUT DROP
```

Engedélyezzük a névfeloldást az első hálózati kártya esetében kívülről:

```
iptables -A INPUT -p udp -i
➥eth0 --sport 53 -j ACCEPT
```

Megtiltjuk, hogy az első hálózati kártya ping utasításokra reagáljon:

```
iptables -A INPUT -p icmp -d
➥-I eth0 -j DROP
```

És így tovább. Legutolsó utasításunkhoz hasonló eredményt kapunk, ha az *IPCop* grafikus felületén a *Tűzfal* fül alatti *Firewall options* -t választva beállítjuk az *icmp* válaszok tiltását a külső, vagy akár az összes hálózati kártyára vonatkoztatva, s ezáltal kívülről „láthatatlanná” tehetjük a tűzfalat, feltéve, hogy nem használunk olyan programot, mely igényli a *ping* utasításra visszaküldendő pozitív válaszok meglétét.
A csomagszűrés jó beállításaival elkerülhető a túlterheléses támadás, valamint sok manipuláció. Akit érdekel az *iptables* megoldások működése, a ➲ *iptables-tutorial.frozentux.net* oldalon sok hasznos információt gyűjthet e témában. Ritkán ugyan, de előfordulhat, hogy kézzel kell szerkesztenünk szabályokat – ilyen szituáció lehet a virtuális magánhálózat kialakítása – de az esetek túlnyomó többségében nem kell hozzányúlnunk beállításaihoz. Az *IPCop* –ot jól felvértezték, s ha külső program nem rontja el az átgondolt láncolatok sorozatát, erős védelmezőnk lehet.

Ne hagyjuk leomlani a falat

Nagyon sok támadási kísérlet nem hozzáértőktől származik, hanem unatkozó, különböző sebeshetőségek után kutató programokat futtató emberektől, akik többnyire csak rövid szkriptprogramok elindításával próbálnak rést találni a rendszereken. Ellenük sokszor már az is segíthet, ha nem szokványos beállítá-

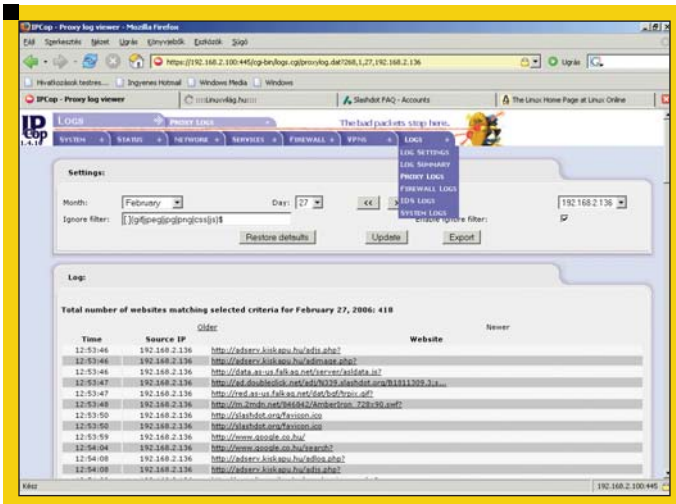
```
Starting nmap 3.81 ( http://www.insecure.org/nmap/ ) at 2006-02-26 17:37 CET
Initiating SYN Stealth Scan against ipcop.localhost (192.168.1.1) [1663 ports] a
t 17:37
Discovered open port 53/tcp on 192.168.1.1
Discovered open port 80/tcp on 192.168.1.1
Discovered open port 800/tcp on 192.168.1.1
Discovered open port 81/tcp on 192.168.1.1
Discovered open port 222/tcp on 192.168.1.1
Discovered open port 445/tcp on 192.168.1.1
The SYN Stealth Scan took 0.24s to scan 1663 total ports.
Host ipcop.localhost (192.168.2.100) appears to be up ... good.
Interesting ports on ipcop.localhost (192.168.2.1):
(The 1657 ports scanned but not shown below are in state: closed)
PORT    STATE SERVICE
53/tcp  open  domain
80/tcp  open  http
81/tcp  open  hosts2-ns
222/tcp open  rsh-spx
445/tcp open  microsoft-ds
800/tcp open  mdbs_daemon
MAC Address: 00:40:AF:53:D1:11 (Cameo Communications)

Nmap finished: 1 IP address (1 host up) scanned in 0.961 seconds
               Raw packets sent: 1665 (66.6KB) | Rcvd: 1664 (76.5KB)
```

1. ábra Nyitott portok után kutatunk

sokat használunk – például ilyen az *ssh* 22-es port helyett a 222-es alkalmazása az *IPCopban* –, nem hagyunk nyitott portokat, és megpróbáljuk naprakészen tartani a rendszert. Egy felkészültebb támadóval szemben ennél azért jóval többre is szükség lehet, így például hosszú és bonyolult jelszavakat használjunk s ahol lehet inkább nyitott kulcsú azonosítást, lehetőség szerint az összes hálózati forgalmat titkosítsuk (például *ssh* alagútba küldve), figyeljük a rendszer állapotát, és a szükséges minimumnál több szolgáltatást ne engedjünk futni, és csak akkor engedélyezzünk távoli hozzáférést, amikor használjuk is. A vírusok, kártevők, trójai programok, kéretlen levélküldözgetők mindkét kategóriába tartoznak, mert egyfelől a mechanikus ismétlődést és gépi intelligenciát könnyebb leleplezni, másfelől egy még fel nem fedezett biztonsági rés ismeretével és új áthatolási módszerekkel felvértezve akár belső hálózatunkról kifelé is próbálkozhatnak jutni, s adatokat továbbítani.
Sok biztonsági rés, puffertúlcsordulásos támadás leírását megtalálhatjuk a ➲ *packetstorm.linuxsecurity.com* honlapon. Utóbbiak ellen részleges védelmet jelenthetnek majd az újabb ezirányú hardverfejlesztések, illetve a jól megválasztott, frissített programcsomagok.
Csábíthat minket például az a tény, hogy egy rugalmasságáról híres *Linux* rendszer fut a hardveren, amelyre akár *Cups*, *Samba* stb. csomagot is telepíthetünk (higgyük el, van aki megteszi) belső hálózatunk fájl- és nyomtatószerver igényeinek kielégítésére – ha nincs más szabad szerverünk, ám óvakodjunk ezektől a látszatmegoldásoktól, mert minden egyes programmal nő a hibalehetőség és a javítatlan rések előfordulási esélye.
Az *IPCop* rendelkezik egy beépített betörésdetektáló szolgáltatással, mely a közismert *Snort* program egy verzióját tartalmazza. Bekapcsolni a *Szolgáltatások* fül *Behatolás Figyelés* menüpontban lehet. Nem árt tisztában lenni azzal, hogy ez általában nem a támadáselhárítást, hanem a megelőzést s az utánkövetést hivatott szolgálni. Ha figyelemmel kísérjük a betörési kísérleteket, talán még idejében be tudjuk zárni a kapukat a hívatlan látogatók előtt.
Nap mint nap készülnek új biztonsági résekről, próbálkozásokról szóló leírások, hibajavítások és sok egyéb, e témát érintő fejlesztés. Ha térítésmentesen regisztráljuk magunkat a ➲ *www.snort.org* honlapon s a kapott ún. *Oink* kódot beírjuk a grafikus felület megfelelő rovatába, lehetőségünk nyílik a program fejlesztőitől a tűzfalon keresztül megkapni a legújabb szabályrendszereket, s ezáltal a vakriasztások számát csökkenteni, a potenciális támadásokat pedig pontosabban előrejelezni. Jellemző lehet például, ha *Naplók* fül *BFR naplók* menüpontját böngészve egy rövid időközön belül ugyanarról a címről több portszámot próbálgatva gyanúra okot adó, naplózott tevékenységgel találkozunk, ilyenkor lehetséges, hogy a nyitott portok után kutat valaki vagy valami.
Ugyanígy leleplezhetjük azon kártevő programokat is, melyek kétségbeesetten próbálnának kijutni a rendszerből

© Kiskapu Kft. Minden jog fenntartva

© Kiskapu Kft. Minden jog fenntartva



2. ábra Vajon mi érdekelte ezt a felhasználót?

s fertőzni más gépeket is, de a tűzfal zárt ajtajairól visszapattannak.
Ha rákattintunk a ***BFR napló*** bejegyzései között található *IP* címekre, a program megpróbálja kinyomozni, kit is takarhat az adott cím. Ez persze nem jelenti azt, hogy a rosszindulatú próbálkozó az adott gép gazdája, lehet egy nyilvános ***proxyszerver*** vagy egy fertőzött gép is. Ha a ***SID:*** felirat melletti azonosítószámra kattintunk, a ***Snort*** honlapjára ugorva részletesebb leírást láthatunk az adott bejegyzésről. Érdemes néha kipróbálni a rendszert kívülről magunknak is: ha van lehetőségünk egy másik ***Linuxot*** futtató gépről ***nmap*** programot használni (legegyszerűbben `nmap -vv 192.168.1.1` feltéve, hogy előbbi a szerver belső hálózati címe, és a másik géppel látják egymást), megtudhatjuk, milyen nyitott portok találhatók a rendszerben. Ugyanezt elméletileg megtehetjük a külső (***RED*** interfész) *IP* cím végigpásztázásával is, de ezt már akár aggályosnak találhatja internetszolgáltatónk, mivel ő nem láthatja szándékainkat, csak egy ***portscan*** kísérlet nyomait.
Betörésdetektálásról angol nyelven olvashatunk például a ➲ *www.sans.org/resources/idfaq/* címen, magyarul pedig érdemes utánanézni a ➲ *wigwam.sztaki.hu* oldalán. Elszántabbak akár csapdát is állíthatnak a nem kívánt látogatóknak, összekapcsolva a tűzfalat egy külső betörésdetektáló rendszerrel: ilyen megoldásról olvashatunk a ➲ *hogwash.sourceforge.net/docs/baitnswitch.html* címen.

Barát és ellenfél a falon belül

Ha nem elég számunkra a csomagszűrés által nyújtott védelem és korlátozás, s szeretnénk gyorsítani valamelyest a böngészésen is, akkor eljött az ideje a ***proxyszerver*** funkció aktiválásának. Mint minden ilyen szolgáltatást, grafikus felületen ezt is a ***Szolgáltatások*** fül alatt érjük el. Ha kipipáljuk a ***Átlátszó Greent***, akkor egy úgynevezett transzparens proxyt üzemelünk be, mely a kliensek átállítása nélkül, számukra gyakorlatilag láthatatlanul üzemel. Így ők nem közvetlenül az interneten található webszerverrel, hanem a közbeiktatott proxyszolgáltatással tartják a kapcsolatot – az ***Upstream proxy*** választása esetén a proxyszervereket lehet egymással, például az internetszolgáltatóéval összefűzni –, ezáltal beazonosíthatóbbá válik a hálózati forgalom, és a sűrűn látogatott oldalak gyorsítótárba helyezésével csökkenhet a sávszélességigény s nő a böngészés sebessége.
A ***proxy*** működésének legjobban úgy tudjuk nyomon követni, ha a ***Naplózás bekapcsolva*** opciót kipipáljuk, s lementjük, majd átmegyünk a ***Naplók*** fülre, s a ***Proxy naplók*** menüpont alatt látható az adott napi felhasználásból származó információ. Ha csak egy adott felhasználót (értsd: gépet ill. *IP* címet) szeretnénk „tetten érni”, akkor a jobb felső sarokban található legördíthető menüpontban a ***Minden*** helyett válasszuk ki a megtekinteni kívánt címet.
A ***proxy*** beállítása után – ha nem akarunk kiegészítő programokat letölteni – legegyszerűbben az hozzáférést szabályzó úgynevezett ***ACL***-listák szerkesztésével tudjuk szűrni a látogatható tartalmakat. Alkalmazhatunk hasonló mechanizmust, mint például sok levelezőszerver: felveszünk feketelistás és fehérlistás felhasználókat illetve felhasználási módokat, s míg előbbiek teljes tiltást kapnak, utóbbiak korlátlan hozzáférést élvezhetnek, s mindezt fűszerezhetjük az éjszakai használat tilalmával. Ennek menete a következő:
Lépjünk be ***ssh***-val az ***IPCopba***, majd egy szimpatikus könyvtárat kiválasztva – például */usr/share* – hozzunk létre egy `tiltolista.txt` nevezetű fájlt. Minderre legegyszerűbb a ***vi*** szövegszerkesztőt használni, de ha kevésbé vagyunk gyakorlottak, megteszi a rendszeren található ***pico*** is.
A fájlba soronként írjuk bele a tiltani kívánt honlap címét – például a nem kívánt tartalmat biztosítókat, a reklámszervereket, a vírusgyanúsakat stb. – s mentsük el. Ugyanígy írjunk egy `szabadlista.txt` állományt, s ebbe írjuk bele azokat az *IP* címeket (értelemszerűen ennek elsődlegesen a belső hálózatunk megbízható gépeinek címét illik tartalmaznia), s azt is mentsük.
Ezek után írjuk át az ***squid*** (figyeljünk rá, hogy nem az */etc/squid* könyvtárban találjuk) konfigurációs beállításait, azaz a */var/ipcop/proxy/squid.conf*-ot. Szúrjuk be az `acl` kezdetű sorok közé a következőt:

```
acl tiltolistank url regex
➥"/usr/share/tiltolista.txt"
acl szabadlistank url regex
➥"/usr/share/szabadlista.txt"
acl ejszaka time 22:00-05:00
```

Ezzel tudatjuk a rendszerrel, hogy az adott fájlok tartalmát internetes címként olvasva a későbbiekben megfogalmazott szabályt érvényesítse reá nézve, az utolsó bejegyzés arról gondoskodik, hogy éjszaka ne használják a rendszert.
Keressük meg a `http_access` kezdetű sorokat, s oda a következő (a tényleges tiltást illetve engedélyezést ez váltja ki) információt helyezzük el egy új sorban:

```
http_access deny tiltolistank
http_access accept
➥szabadlistank
http_access deny ejszaka
```

Arra is figyelnünk kell, nem történt-e más szabálymeghatározás már korábban, például az általunk tiltólistára vett *IP* címekből nincs-e explicit engedélyezve vagy a megbízhatónak ítéltekből tiltva valamelyik, mert ez esetben legegyszerűbb az ellentmondó szabályokból a nem kívántat törölni. Szerencsénkre több kiegészítő alkalmazás is készült, mely arra hivatott, hogy a fáradságos gépelést kiváltsa, és az opciók kiválasztását grafikus felületű navigációval segítse. Ezen alkalmazások egyike az ***Advproxy*** (➲ *www.advproxy.net*), vagy például a *Cop+* csomag (➲ *home.earthlink.net/~copplus/*) is hathatós segítséget tud nyújtani a *proxyszerver* felhasználóbarát módon történő testre szabásához. Néha még a csomagszűrés és *proxy* együttes használata is kevésnek bizonyulhat. Az egyenrangú hálózatként ismertté vált fájlcserélő szolgáltatások (úgynevezett *p2p* alkalmazások) térhódításával új kihívások jelentek meg egyes rendszergazdák mindennapjaiban. Ezen alkalmazások jól tudnak alkalmazkodni heterogén környezetekhez, portszámot és *IP* címet tudnak váltogatni, s tevékenységüket akár másnak álcázva is képesek működni. Ellenük megoldást jelenthetnek az alkalmazásszintű szűrők, egyikük az ➲ *l7-filter.sourceforge.net* honlapról tölthető le.

Havi nagytakarítás

Saját magunknak is okozhatunk fejfájást, ha például hagyjuk a naplózásból származó fájlokat felszaporodni, ennek megoldására használható például a `logrotate` program (ciklikussá teszi a naplófájlokat, azaz törli a megadott időpontnál régebbi bejegyzéseket). Aktuális állapotuk az *IPCop* disztribúciókban egy jól áttekinthető felületről nyomon követhető, csakúgy, mint a rendszer egyéb fontos erőforrásainak állapota. Így elkerülhető, hogy olyan és rendszerállapot-mérő programokat, mint például a

```
ps aux
netstat -p
du -hs
```

folyamatosan futtatni kényszerüljünk, mert ezek kimenetéhez hasonló, de egységes és könnyen értékelhető grafikus és szöveges információkat közöl velünk a rendszer. Ha tartunk tőle, hogy a támadó maga után eltakarítja a betörés nyomait, legjobb, ha a rendszernaplózó `syslog` fájl kimenetét egy másik számítógépre, vagy még inkább nyomtatóra irányítjuk. Így – hacsak nincs fizikai hozzáférése a nyomtatóhoz és géphez – elméletileg minden változtatási kísérlete előtt papírra tudjuk vetni a gyanús eseményeket. Ne bízzuk el teljesen magunkat, a nyomtatósor átirányításával, vagy egy másik biztonsági rés kihasználásával sajnos némi esélye még marad ennek a megakadályozására is.

Biztos hogy én építettem ezt a falat?

A rendszer konzisztenciájának vizsgálatára részben segítséget kapunk az *IPCop* grafikus felületén a frissítések letöltésekor, tehát elméletileg csekély az esélye annak, hogy az újabb hivatalos kiadások használatakor tesszük sebezhetőbbé a tűzfalat. Ugyanakkor az interneten böngészve találhatunk külső fejlesztők által írt, hasznosnak ígérkező kiegészítőket, melyeket általában ezt az ellenőrzést megkerülve vagyunk kénytelenek telepíteni. Különböző segédprogramok és szoftvercsomagok állnak rendelkezésre a rendszerek védelmének megerősítésére és vizsgálatára. A fájlok módosítását például a *Tripwire* nevű alkalmazás segítségével tudnánk megnézni, de ha nem akarunk a telepítésével bajlódni, akkor egyszerűen kiadhatjuk parancssorból az

```
md5sum * > ellenorzo
```

utasítást, mellyel az aktuális könyvtárban található fájlokról készítünk egy ún. ellenőrző összeget (***md5 hash*** néven is találkozhatunk az eljárással) az `ellenorzo` nevű fájlba. Ez lehetővé teszi, hogy észrevegyük változásukat – például egy szándékos rosszindulatú módosítást.
Ellenőrzéshez csak ezt generált fájlt kell összevetni a könyvtárban találhatókkal:

```
md5sum –check ellenorzo | grep
➥FAILED
```

Ha problémát talál kiírja. Fontos lehet még, hogy az utolsó ellenőrzőösszeg generálása óta keletkeztek-e a könyvtárban újabb fájlok (természetesen első futtatás alkalmával az `ellenorzo` fájl létrehozása miatt eggyel több a darabszám!); ezt egyszerűen a következő héjprogram segítségével tudhatjuk meg:

```
most=`ls -l | wc -l`
korabban=`md5sum -check
➥ellenorzo | wc -l`
if test $most -gt $korabban
then echo "Ennyivel több darab
➥fájl van most a
➥könyvtárunkban:"
expr $most - $korabban
fi
```

Nekünk kell eldöntenünk, mely könyvtárakat érdemes az ellenőrzésbe bevonni, s ezután akár automatizálhatjuk is egy-két egyszerűbb ***shellszkripttel***, rekurzív könyvtárbejárást alkalmazva a vizsgálatot. Érdemes figyelni a rejtett fájlokra (például `ls .[!.]*` paranccsal kiszűrhetők és elkülöníthetők a szintén ponttal kezdődő könyvtárnevektől), a *swapfile* kihagyására, esetlegesen a */tmp* és */proc* átugrására is, viszont a */dev* bevonására is, s nemcsak a keletkező, de az eltünedező fájlok is gyanút kelthetnek bennünk.
Hálózati biztonsággal, tűzfalakkal foglalkozó tanulmányok és jótanácsok bőséges lelőhelye az internet. Hasznos olvasmány a ➲ *www.cert.org/security-improvement* valamint a *wikioldalak* között is fellelhető fontosabb fogalmak magyarázata. Készüljünk fel rá, hogy sokszor egymásnak ellentmondó tanácsokat és elveket fogunk olvasni. Általában saját magunknak kell kitalálnunk az üdvözítő megoldást, mivel legjobban mi ismerjük a védeni kívánt rendszerek sajátosságait, de sokat segíthet egy olyan „kulcsrakész” rendszer, mint az *IPCop*.

Tóth Virgil Zoltán
(m_v@c2.hu)
Szoftverfejlesztő informatikus és rendszergazda, kedvence a Debian disztribúció. Szabadidejét legszívesebben felesége és szépirodalmi regények társaságában tölti. Lenyűgözőnek tartja a Linux rugalmasságát, és a vele dolgozók aktivitását.

© Kiskapu Kft. Minden jog fenntartva